Group Meeting
(MN2L)

10/05/2021

1. Time integration in PFEM

2. Results comparison:
Backward-Euler & α-Method

3. Explicit time integration scheme used in PFEM-2

10/05/2021

1. Time integration in PFEM

2. Results comparison:
Backward-Euler & α-Method

3. Explicit time integration scheme
used in PFEM-2

10/05/2021

Time Integration in PFEM

Motivation

- Understand the different time integration schemes.
- Study the influence of time integration methods in PFEM.

Some time integration schemes in dynamic problems

Linear momentum equation

$$\boxed{\mathbf{M}\,\dot{\mathbf{v}} + \mathbf{K}\,\mathbf{v} - \mathbf{D}^{\intercal}\mathbf{p} = \mathbf{f}^{\mathrm{ext}}}$$

PFEM Framework:

- Incompressible Newtonian Fluid
- PSPG stabilization
- Piccard Algorithm

Some time integration schemes in dynamic problems

Linear momentum equation

$$\boxed{\mathbf{M}\,\dot{\mathbf{v}} + \mathbf{K}\,\mathbf{v} - \mathbf{D}^{\intercal}\mathbf{p} = \mathbf{f}^{\mathrm{ext}}}$$

Where : $$\frac{\mathrm{d}\mathbf{v}(\mathbf{x}^{\mathrm{p}})}{\mathrm{d}t} = \dot{\mathbf{v}}(t, \mathbf{x}^{\mathrm{p}})$$

integration : $$\int \mathrm{d}\mathbf{v} = \int \dot{\mathbf{v}}(t, \mathbf{x}^{\mathrm{p}})\mathrm{d}t$$

Time discret. : $$\mathbf{v}_{\mathrm{n}+1}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}+1}) - \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) = \int_{t_{\mathrm{n}}}^{t_{\mathrm{n}+1}} \dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau$$

Time step : $$\Delta t = t_{\mathrm{n}+1} - t_{\mathrm{n}}$$

Some time integration schemes in dynamic problems

Linear momentum equation

$$\boxed{\mathbf{M}\,\dot{\mathbf{v}} + \mathbf{K}\,\mathbf{v} - \mathbf{D}^{\intercal}\mathbf{p} = \mathbf{f}^{\mathrm{ext}}}$$

Where : $$\frac{\mathrm{d}\mathbf{x}^{\mathrm{p}}}{\mathrm{d}t} = \mathbf{v}(t, \mathbf{x}^{\mathrm{p}})$$

integration : $$\int \mathrm{d}\mathbf{x} = \int \mathbf{v}(t, \mathbf{x}^{\mathrm{p}})\mathrm{d}t$$

Time discret. : $$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} - \mathbf{x}^{\mathrm{p}}_{n} = \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau$$

Time step : $$\Delta t = t_{\mathrm{n+1}} - t_{\mathrm{n}}$$

Some time integration schemes in dynamic problems

Exact expression for integrating state variables:

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{n} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau$$

$\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ and $\mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ are unknown time- and space-dependent functions.

$\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ and $\mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ need to be approximated. (Accuracy will depend on the accuracy of the approximation function, on the time and space discretization).

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.a) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) = \dot{\mathbf{v}}_{\mathrm{n}}$ *(Explicit)*

Integrating the acceleration:

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}_{\mathrm{n}} \mathrm{d}\tau$$

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \Delta t \; \dot{\mathbf{v}}_{\mathrm{n}}}$$

(Forward Euler / Runge-Kutta)

Integrating the velocity function:

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) \mathrm{d}\tau$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) + (\tau - t_{\mathrm{n}}) \dot{\mathbf{v}}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) \mathrm{d}\tau$$

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \; \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) + \frac{\Delta t^2}{2} \dot{\mathbf{v}}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}})}$$

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.b) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n+1}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \dot{\mathbf{v}}_{\mathrm{n+1}}$ *(Implicit)*

Integrating the acceleration:

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}_{\mathrm{n+1}} \mathrm{d}\tau$$

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \Delta t \; \dot{\mathbf{v}}_{\mathrm{n+1}}}$$

(Backward Euler)

Integrating the velocity function:

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) \mathrm{d}\tau$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) + (\tau - t_{\mathrm{n}}) \dot{\mathbf{v}}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) \mathrm{d}\tau$$

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \; \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) + \frac{\Delta t^2}{2} \dot{\mathbf{v}}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}})}$$

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.c) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n+1}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \dot{\mathbf{v}}_{\mathrm{n+1}}$ *(Implicit)*

$$\mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \mathbf{v}(t_{\mathrm{n+1}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n+1}}$$

Integrating the acceleration:

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}_{\mathrm{n+1}} \mathrm{d}\tau$$

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \Delta t \, \dot{\mathbf{v}}_{\mathrm{n+1}}}$$

(Backward Euler)

Integrating the velocity function:

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) \mathrm{d}\tau$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) \mathrm{d}\tau$$

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \, \mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}})}$$

(Backward Euler)

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.a) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) = \dot{\mathbf{v}}_{\mathrm{n}}$

(Explicit)

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \Delta t \; \dot{\mathbf{v}}_{\mathrm{n}}}$$

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \; \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) + \frac{\Delta t^2}{2} \dot{\mathbf{v}}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}})}$$

1.b) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n+1}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \dot{\mathbf{v}}_{\mathrm{n+1}}$

(Implicit)

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \Delta t \; \dot{\mathbf{v}}_{\mathrm{n+1}}}$$

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \; \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}}) + \frac{\Delta t^2}{2} \dot{\mathbf{v}}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}})}$$

1.c) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n+1}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \dot{\mathbf{v}}_{\mathrm{n+1}}$

$$\mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \mathbf{v}(t_{\mathrm{n+1}}, \mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n+1}}$$

(Implicit)

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \Delta t \; \dot{\mathbf{v}}_{\mathrm{n+1}}}$$

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \; \mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}})}$$

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.d) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = (1-\gamma)\dot{\mathbf{v}}_{\mathrm{n}} + \gamma\ \dot{\mathbf{v}}_{\mathrm{n+1}}$

*(Generalized mid-point rule)**

*($\gamma = 1/2$, trapezoidal rule)**

*($\gamma = 0$, Forward Euler)**

*($\gamma = 1$, Backward Euler)**

Integrating the acceleration:

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}} + (1-\gamma)\,\Delta t\ \dot{\mathbf{v}}_{\mathrm{n}} + \gamma\,\Delta t\ \dot{\mathbf{v}}_{\mathrm{n+1}}}$$

Integrating the velocity function:

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \mathbf{v}_{\mathrm{n}} + (1-\gamma)\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n}} + \gamma\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n+1}}}$$

** Lovrić, A., Dettmer, W. G., Kadapa, C., & Perić, D. (2018). A new family of projection schemes for the incompressible Navier–Stokes equations with control of high-frequency damping. CMAME, 339, 160-183.*

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.e) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = (1-\gamma)\dot{\mathbf{v}}_{\mathrm{n}} + \gamma\, \dot{\mathbf{v}}_{\mathrm{n+1}}$ (for velocites)

(Newmark)

$$\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = (1-2\beta)\dot{\mathbf{v}}_{\mathrm{n}} + 2\beta\, \dot{\mathbf{v}}_{\mathrm{n+1}}$$ (for displacements)

Integrating the acceleration:

$$\boxed{\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}} + (1-\gamma)\,\Delta t\; \dot{\mathbf{v}}_{\mathrm{n}} + \gamma\,\Delta t\; \dot{\mathbf{v}}_{\mathrm{n+1}}}$$

Integrating twice the blue acceleration function:

$$\boxed{\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \mathbf{v}_{\mathrm{n}} + (1-2\beta)\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n}} + 2\beta\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n+1}}}$$

1) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ and around $\mathbf{x}^{\mathrm{p}}$

1.f) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = (1-\gamma)\dot{\mathbf{v}}_{\mathrm{n}} + \gamma\, \dot{\mathbf{v}}_{\mathrm{n+1}}$ (for velocites)

(Newmark)

$\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = (1-2\beta)\dot{\mathbf{v}}_{\mathrm{n}} + 2\beta\, \dot{\mathbf{v}}_{\mathrm{n+1}}$ (for displacements)

Weighted or Interpolated state variables:

$$\boxed{\begin{aligned} \dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} &= (1-\alpha_m)\dot{\mathbf{v}}_{\mathrm{n}} + \alpha_m\, \dot{\mathbf{v}}_{\mathrm{n+1}} \\ \mathbf{v}_{\mathrm{n}+\alpha_f} &= (1-\alpha_f)\mathbf{v}_{\mathrm{n}} + \alpha_f\, \mathbf{v}_{\mathrm{n+1}} \end{aligned}}$$

*(Generalized α-Method)**

ρ_∞ : Spectral radius

To be used in:

$$\mathbf{M}\,\dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} + \mathbf{K}\,\mathbf{v}_{\mathrm{n}+\alpha_f} - \mathbf{D}^{\intercal}\mathbf{p} = \mathbf{f}^{\mathrm{ext}}$$

** Lovrić, A., Dettmer, W. G., Kadapa, C., & Perić, D. (2018). A new family of projection schemes for the incompressible Navier–Stokes equations with control of high-frequency damping. CMAME, 339, 160-183.*

Time Integration : Generalized α-Method in PFEM

PFEM Framework:

- Incompressible Newtonian Fluid
- Velocity-Pressure formulation
- PSPG stabilization
- Piccard Algorithm

- Newmark Integration:

$$\mathbf{v}_{\mathrm{n}+1}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}+1}) = \mathbf{v}_{\mathrm{n}} + (1-\gamma)\,\Delta t\;\dot{\mathbf{v}}_{\mathrm{n}} + \gamma\,\Delta t\;\dot{\mathbf{v}}_{\mathrm{n}+1} \tag{1}$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n}+1} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \mathbf{v}_{\mathrm{n}} + (1-2\beta)\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n}} + 2\beta\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n}+1} \tag{2}$$

- α-Method:

$$\dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} = (1-\alpha_m)\dot{\mathbf{v}}_{\mathrm{n}} + \alpha_m\;\dot{\mathbf{v}}_{\mathrm{n}+1} \tag{3}$$

$$\mathbf{v}_{\mathrm{n}+\alpha_f} = (1-\alpha_f)\mathbf{v}_{\mathrm{n}} + \alpha_f\;\mathbf{v}_{\mathrm{n}+1} \tag{4}$$

Time Integration : Generalized α-Method in PFEM

Working (1) :
$$\dot{\mathbf{v}}_{\mathrm{n+1}} = \frac{1}{\gamma\,\Delta t}(\mathbf{v}_{\mathrm{n+1}} - \mathbf{v}_{\mathrm{n}}) - (1/\gamma - 1)\,\dot{\mathbf{v}}_{\mathrm{n}} \tag{5}$$

(5) in (3) :
$$\dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} = \frac{\alpha_m}{\gamma\Delta t}(\mathbf{v}_{\mathrm{n+1}} - \mathbf{v}_{\mathrm{n}}) + (1 - \alpha_m/\gamma)\dot{\mathbf{v}}_{\mathrm{n}} \tag{6}$$

(4) and (6) in :
$$\mathbf{M}\,\dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} + \mathbf{K}\,\mathbf{v}_{\mathrm{n}+\alpha_f} - \mathbf{D}^{\intercal}\mathbf{p} = \mathbf{f}^{\mathrm{ext}}$$

… yields :
$$\begin{aligned}\mathbf{M}\,\frac{\alpha_m}{\gamma\,\Delta t}\mathbf{v}_{\mathrm{n+1}} + \mathbf{K}\,\alpha_f\,\mathbf{v}_{\mathrm{n+1}} - \mathbf{D}^{\intercal}\mathbf{p}_{\mathrm{n+1}} &= \\ \mathbf{f}^{\mathrm{ext}} + \mathbf{M}\left(\frac{\alpha_m}{\gamma\,\Delta t}\mathbf{v}_{\mathrm{n}} - (1-\alpha_m/\gamma)\dot{\mathbf{v}}_{\mathrm{n}}\right) &+ \mathbf{K}\,(\alpha_f - 1)\,\mathbf{v}_{\mathrm{n}}\end{aligned} \tag{7}$$

(In blue the added terms w.r.t. Backward Euler integration scheme)

Time Integration : Generalized α-Method in PFEM

PFEM Framework:

- Incompressible Newtonian Fluid
- **Position-Pressure formulation**
- PSPG stabilization
- Piccard Algorithm

- Newmark Integration:

$$\mathbf{v}_{\mathrm{n}+1}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}+1}) = \mathbf{v}_{\mathrm{n}} + (1-\gamma)\,\Delta t\ \dot{\mathbf{v}}_{\mathrm{n}} + \gamma\,\Delta t\ \dot{\mathbf{v}}_{\mathrm{n}+1} \qquad (1)$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n}+1} = \mathbf{x}^{\mathrm{p}}_{\mathrm{n}} + \Delta t \mathbf{v}_{\mathrm{n}} + (1-2\beta)\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n}} + 2\beta\frac{\Delta t^2}{2}\dot{\mathbf{v}}_{\mathrm{n}+1} \qquad (2)$$

- α-Method:

$$\dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} = (1-\alpha_m)\dot{\mathbf{v}}_{\mathrm{n}} + \alpha_m\ \dot{\mathbf{v}}_{\mathrm{n}+1} \qquad (3)$$

$$\mathbf{v}_{\mathrm{n}+\alpha_f} = (1-\alpha_f)\mathbf{v}_{\mathrm{n}} + \alpha_f\ \mathbf{v}_{\mathrm{n}+1} \qquad (4)$$

Time Integration : Generalized α-Method in PFEM

from (2) : $$\dot{\mathbf{v}}_{\mathrm{n}+1} = \frac{1}{\beta\,\Delta t^2}(\mathbf{x}_{\mathrm{n}+1} - \mathbf{x}_{\mathrm{n}}) - \frac{1}{\beta\,\Delta t}\,\mathbf{v}_{\mathrm{n}} + \left(1 - \frac{1}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}} \tag{5}$$

(5) in (1) : $$\mathbf{v}_{\mathrm{n}+1} = \frac{\gamma}{\beta\,\Delta t}(\mathbf{x}_{\mathrm{n}+1} - \mathbf{x}_{\mathrm{n}}) + \frac{\beta - \gamma}{\beta}\,\mathbf{v}_{\mathrm{n}} + \left(\frac{2\beta - \gamma}{2\beta}\right)\Delta t\,\dot{\mathbf{v}}_{\mathrm{n}} \tag{6}$$

(5) in (3) : $$\dot{\mathbf{v}}_{\mathrm{n}+\alpha_m} = \frac{\alpha_m}{\beta\Delta t^2}(\mathbf{x}_{\mathrm{n}+1} - \mathbf{x}_{\mathrm{n}}) - \frac{\alpha_m}{\beta\Delta t}\mathbf{v}_{\mathrm{n}} + \left(1 - \frac{\alpha_m}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}} \tag{7}$$

(6) in (4) :

$$\mathbf{v}_{\mathrm{n}+\alpha_f} = \frac{\alpha_f\gamma}{\beta\Delta t}(\mathbf{x}_{\mathrm{n}+1} - \mathbf{x}_{\mathrm{n}}) + \left(1 - \frac{\alpha_f\gamma}{\beta}\right)\mathbf{v}_{\mathrm{n}} + \alpha_f\Delta t\left(1 - \frac{\gamma}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}} \tag{8}$$

Time Integration : Generalized α-Method in PFEM

... yields :

$$\mathbf{M}\,\frac{\alpha_m}{\beta\,\Delta t^2}(\mathbf{x}_{\mathrm{n+1}}-\mathbf{x}_{\mathrm{n}})+\mathbf{K}\,\frac{\alpha_f\gamma}{\beta\Delta t}(\mathbf{x}_{\mathrm{n+1}}-\mathbf{x}_{\mathrm{n}})-\mathbf{D}^{\mathsf{T}}\mathbf{p}_{\mathrm{n+1}}= \tag{9}$$

$$\mathbf{f}^{\mathrm{ext}}+\mathbf{M}\left(\frac{\alpha_m}{\beta\,\Delta t}\mathbf{v}_{\mathrm{n}}-\left(1-\frac{\alpha_m}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}}\right)+\mathbf{K}\left(\left(\frac{\gamma\alpha_f}{\beta}-1\right)\mathbf{v}_{\mathrm{n}}-\alpha_f\Delta t\left(1-\frac{\gamma}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}}\right)$$

(In blue the added terms w.r.t. Backward Euler integration scheme)
(In red the added terms w.r.t. α-Method integration scheme in a velocity-pressure formulation)

Time Integration : Generalized α-Method in PFEM

Summary

Backward-Euler & Velocity-Pressure

$$\mathbf{M}\,\frac{1}{\Delta t}\mathbf{v}_{\mathrm{n+1}} + \mathbf{K}\,\mathbf{v}_{\mathrm{n+1}} - \mathbf{D}^{\mathsf{T}}\mathbf{p}_{\mathrm{n+1}} = \mathbf{f}^{\mathrm{ext}} + \mathbf{M}\,\frac{1}{\Delta t}\mathbf{v}_{\mathrm{n}}$$

α-Method & Velocity-Pressure

$$\mathbf{M}\,\frac{\alpha_m}{\gamma\,\Delta t}\mathbf{v}_{\mathrm{n+1}} + \mathbf{K}\,\alpha_f\,\mathbf{v}_{\mathrm{n+1}} - \mathbf{D}^{\mathsf{T}}\mathbf{p}_{\mathrm{n+1}} =$$

$$\mathbf{f}^{\mathrm{ext}} + \mathbf{M}\left(\frac{\alpha_m}{\gamma\,\Delta t}\mathbf{v}_{\mathrm{n}} - (1-\alpha_m/\gamma)\dot{\mathbf{v}}_{\mathrm{n}}\right) + \mathbf{K}\,(\alpha_f - 1)\,\mathbf{v}_{\mathrm{n}}$$

α-Method & Displacement-Pressure

$$\mathbf{M}\,\frac{\alpha_m}{\beta\,\Delta t^2}(\mathbf{x}_{\mathrm{n+1}} - \mathbf{x}_{\mathrm{n}}) + \mathbf{K}\,\frac{\alpha_f\gamma}{\beta\Delta t}\,(\mathbf{x}_{\mathrm{n+1}} - \mathbf{x}_{\mathrm{n}}) - \mathbf{D}^{\mathsf{T}}\mathbf{p}_{\mathrm{n+1}} =$$

$$\mathbf{f}^{\mathrm{ext}} + \mathbf{M}\left(\frac{\alpha_m}{\beta\,\Delta t}\mathbf{v}_{\mathrm{n}} - \left(1 - \frac{\alpha_m}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}}\right) + \mathbf{K}\left(\left(\frac{\gamma\alpha_f}{\beta} - 1\right)\mathbf{v}_{\mathrm{n}} - \alpha_f\Delta t\left(1 - \frac{\gamma}{2\beta}\right)\dot{\mathbf{v}}_{\mathrm{n}}\right)$$

Nonlinear algorithm for Velocity-Pressure formulation

Backward-Euler

Known: $\mathbf{x}_\mathrm{n},\ \mathbf{v}_\mathrm{n},\ \dot{\mathbf{v}}_\mathrm{n}$

1) Set k = 0
2) Initial guess: $\mathbf{x}_k = \mathbf{x}_\mathrm{n}$
3) *While* (convergence is not reached)
   - 3.1) Compute:
     $$\mathbf{A}_\mathrm{BE}(\mathbf{x}_k, \mathbf{v}_k)$$
     $$\mathbf{b}_\mathrm{BE}(\mathbf{x}_k, \mathbf{v}_\mathrm{n}, \mathbf{v}_\mathrm{k})$$
   - 3.2) Solve : $\mathbf{A}_\mathrm{BE}\,\mathbf{q}_{\mathrm{n}+1} = \mathbf{b}_\mathrm{BE}$
   - 3.3) Update : $\mathbf{x}_k = \mathbf{x}_\mathrm{n} + \Delta t\ \mathbf{v}_{k+1}$
   - 3.4) Set : $k = k + 1$
4) Set : $\mathbf{q}_{\mathrm{n}+1} = \mathbf{q}_\mathrm{k}$

   Update : $\mathbf{x}_{\mathrm{n}+1} = \mathbf{x}_\mathrm{n} + \Delta t\ \mathbf{v}_{\mathrm{n}+1}$
5) RE-MESH (if necessary)

α-Method

Known: $\mathbf{x}_\mathrm{n},\ \mathbf{v}_\mathrm{n},\ \dot{\mathbf{v}}_\mathrm{n}$

1) Set k = 0
2) Initial guess: $\mathbf{x}_k = \mathbf{x}_\mathrm{n}$
3) *While* (convergence is not reached)
   - 3.1) Compute:
     $$\mathbf{A}_\mathrm{AM}(\mathbf{x}_k, \mathbf{v}_k, \gamma, \beta, \alpha_m, \alpha_f)$$
     $$\mathbf{b}_\mathrm{AM}(\mathbf{x}_k, \mathbf{v}_k, \mathbf{v}_\mathrm{n}, \dot{\mathbf{v}}_\mathrm{n}, \gamma, \beta, \alpha_m, \alpha_f)$$
   - 3.2) Solve : $\mathbf{A}_\mathrm{AM}\,\mathbf{q}_{\mathrm{n}+1} = \mathbf{b}_\mathrm{AM}$
   - 3.3) Update : $\mathbf{x}_k$ using Newmark
   - 3.4) Set : $k = k + 1$
4) Set : $\mathbf{q}_{\mathrm{n}+1} = \mathbf{q}_\mathrm{k}$

   Update : $\mathbf{x}_{\mathrm{n}+1}$ using Newmark
5) RE-MESH (if necessary)

1. Time integration in PFEM

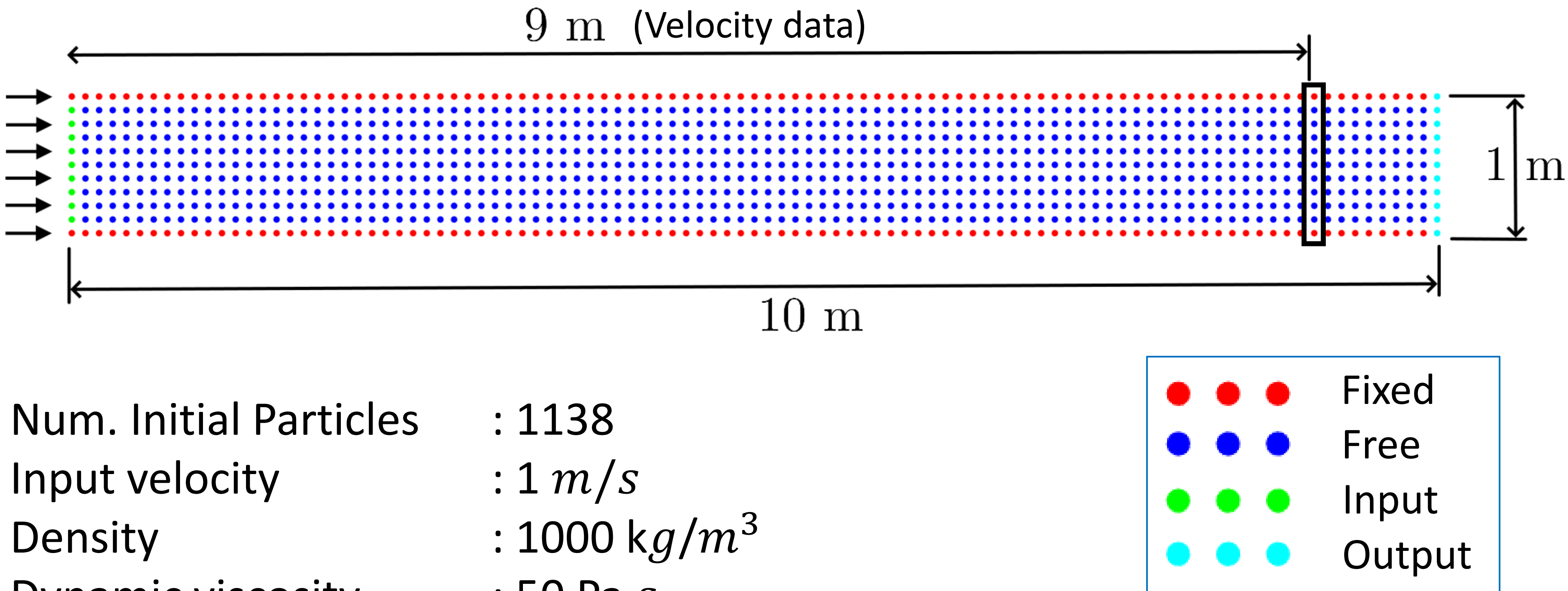
2. Results comparison: Backward-Euler & α-Method

3. Explicit time integration scheme used in PFEM-2

10/05/2021

Comparison of time integration schemes

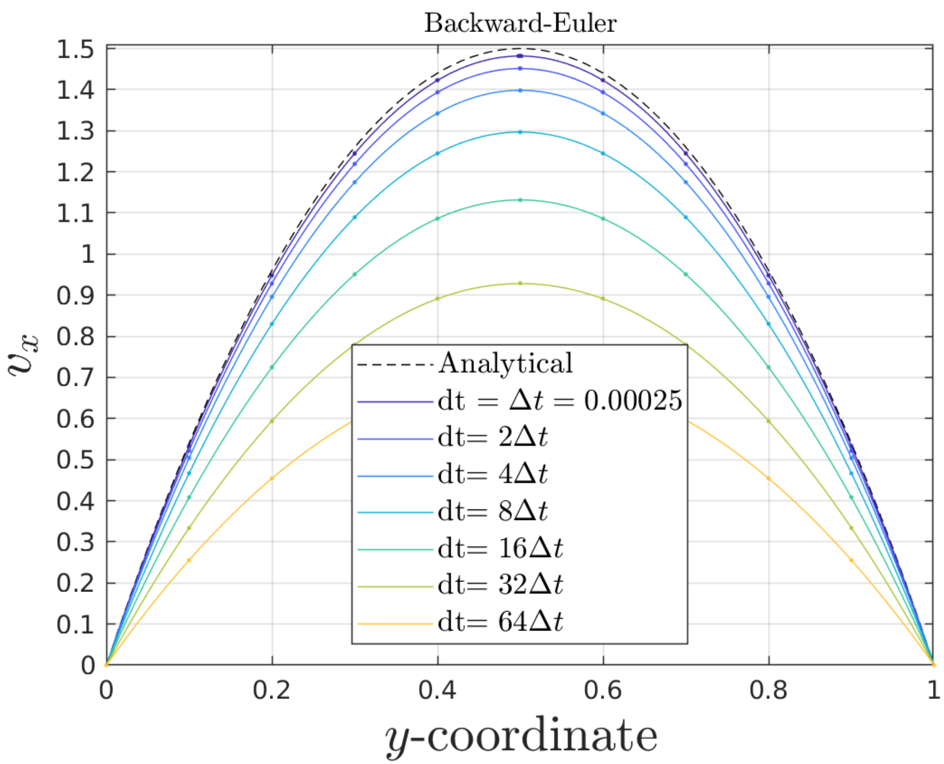
Test case : Flow between 2 plates.



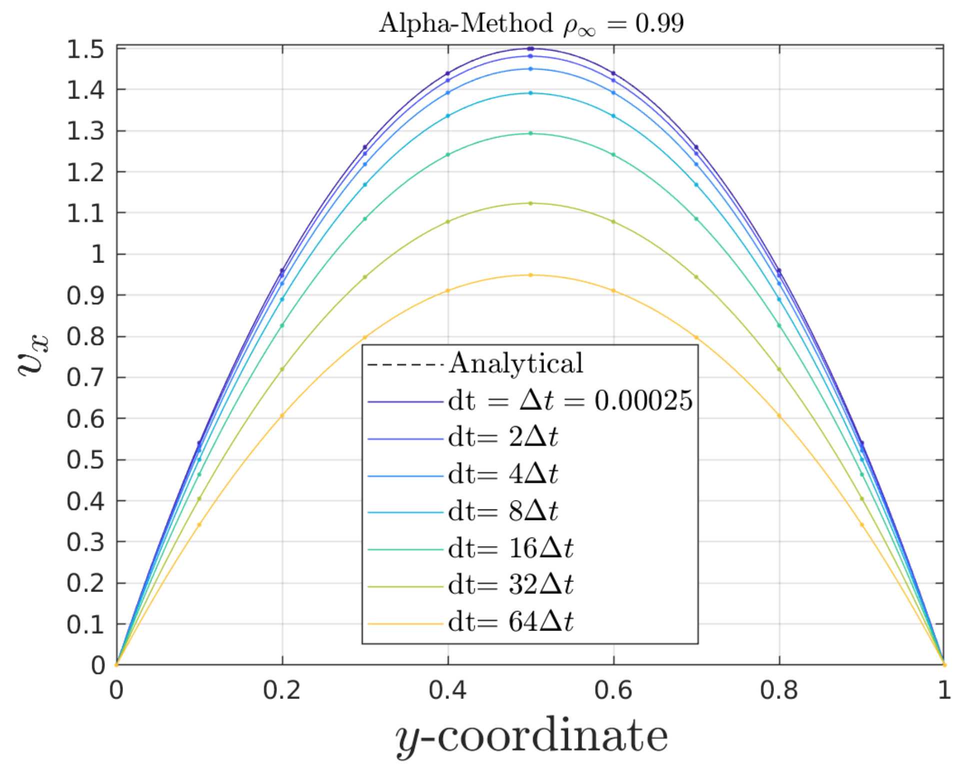
Num. Initial Particles	: 1138
Input velocity	: 1 m/s
Density	: 1000 kg/m^3
Dynamic viscosity	: 50 Pa s.
Simulation span	: 5 seconds
Initial time step	: 0.0001
Maximum time step	: 0.00025 , 0.0005, 0.001, 0.002, 0.004, 0.008, 0.016
(7 cases)	: [Δt , $2\Delta t$, $4\Delta t$, $8\,\Delta t$, $16\Delta t$, $32\Delta t$, $64\Delta t$]

Flow Between 2 plates : Results (BE and αM)

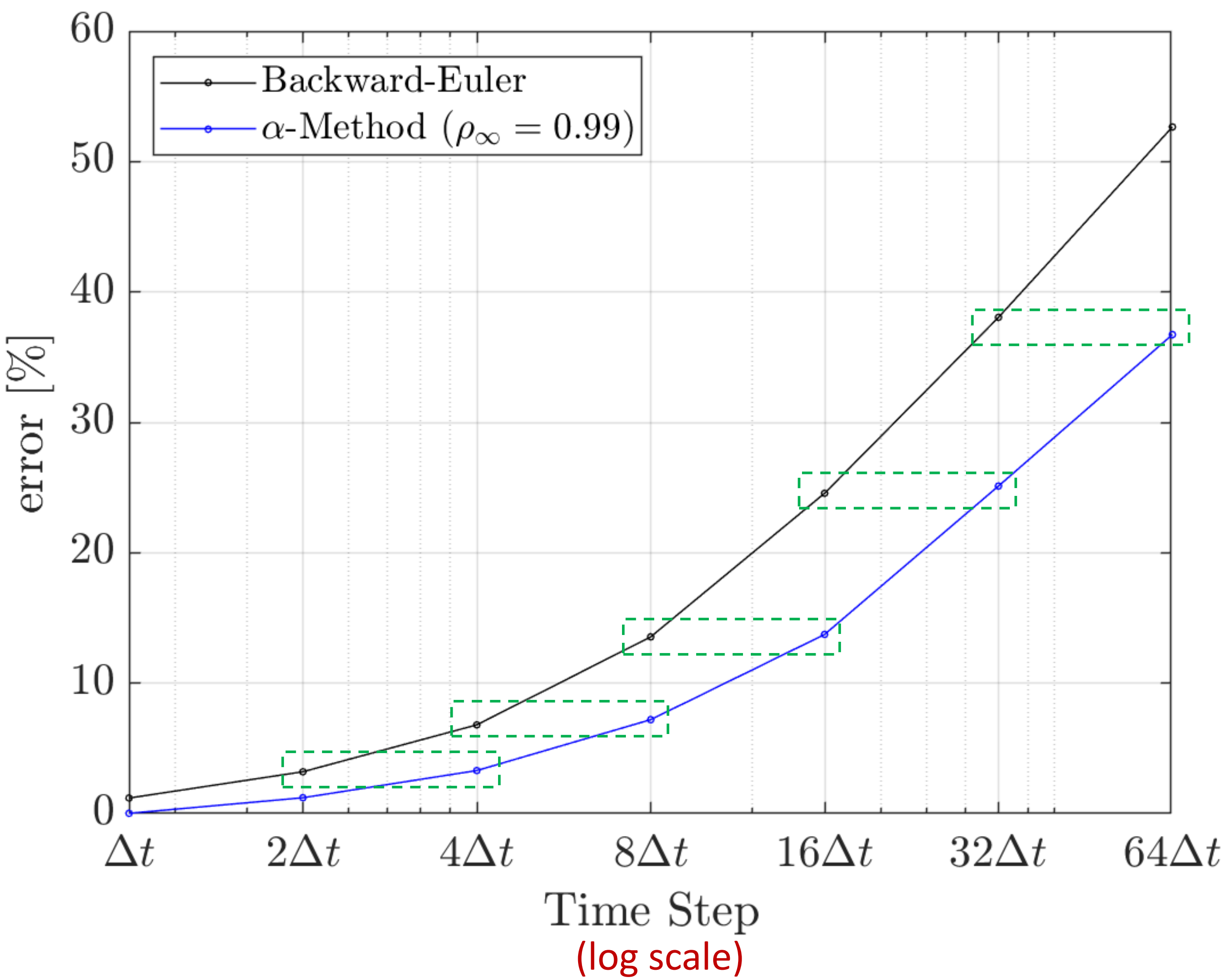
<u>Backward-Euler</u>



<u>α-Method</u>



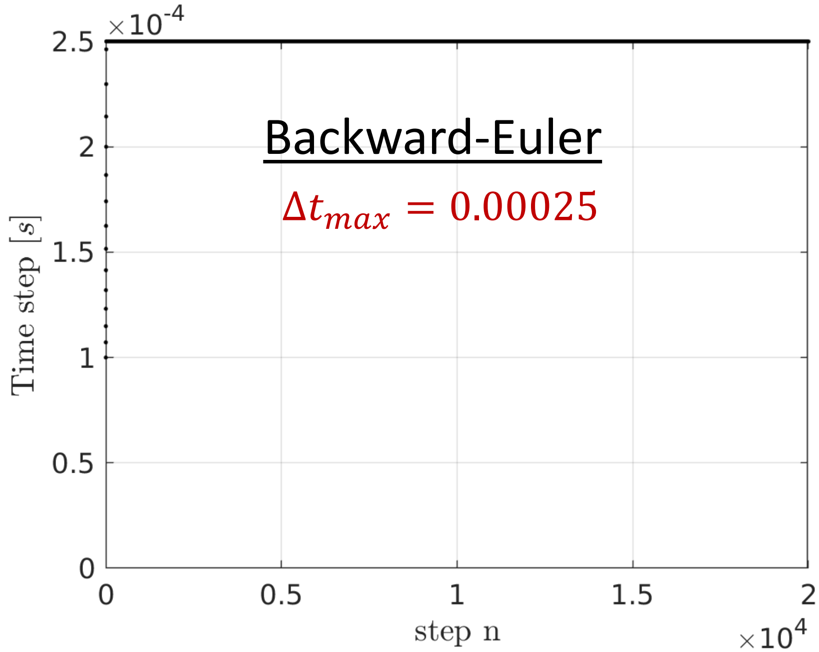
Flow Between 2 plates : Results (BE and αM)

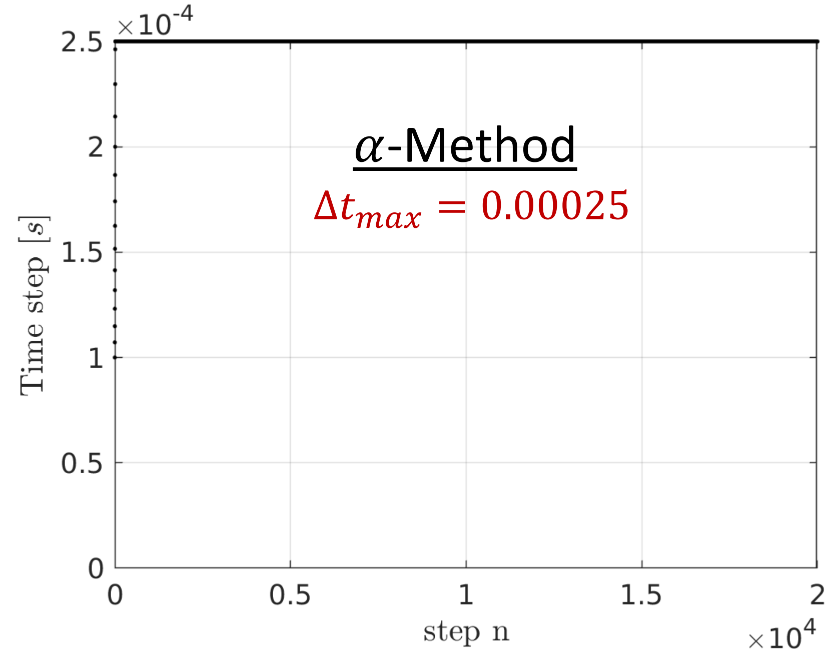


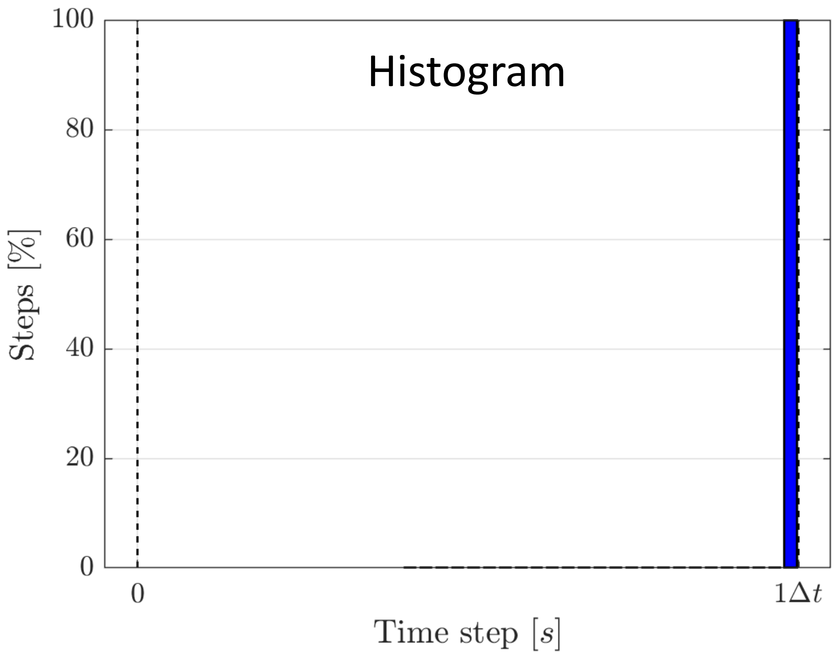
Generally, in this test case, α -Method matches the accuracy of Backward-Euler even using twice the time step of Backward-Euler.

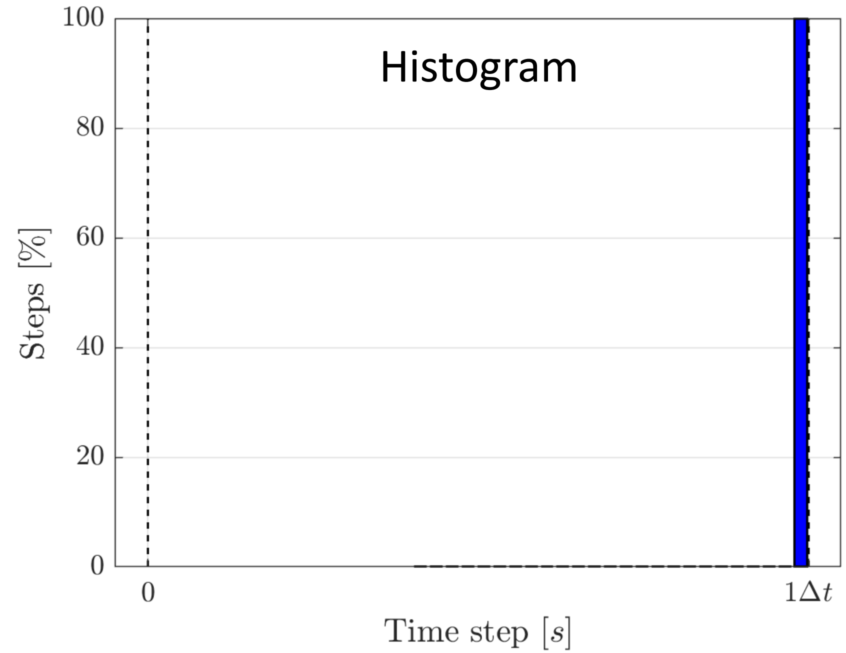
$$\text{error} = \frac{|\max(\mathbf{v_x}) - 1.5|}{1.5} \times 100$$

Flow Between 2 plates : Results (BE and αM)

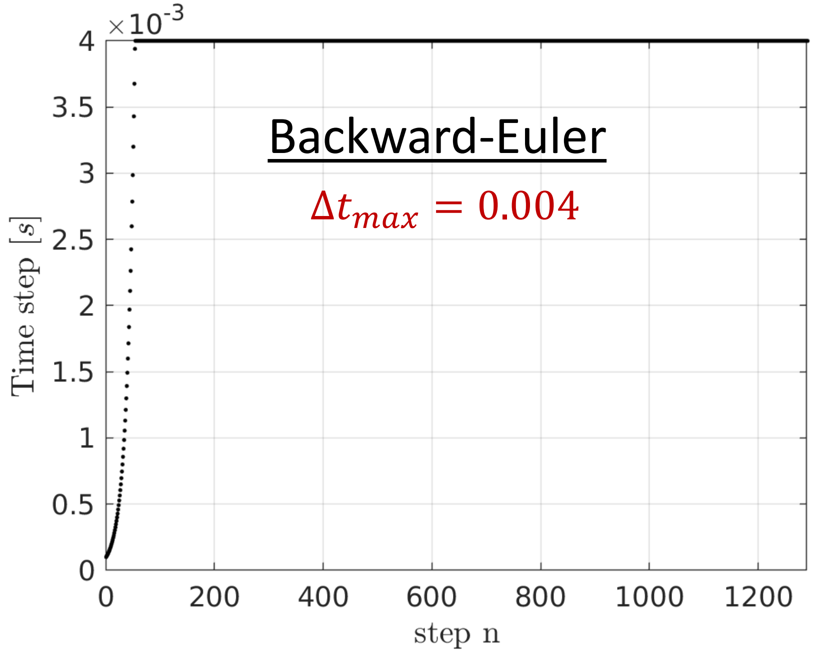


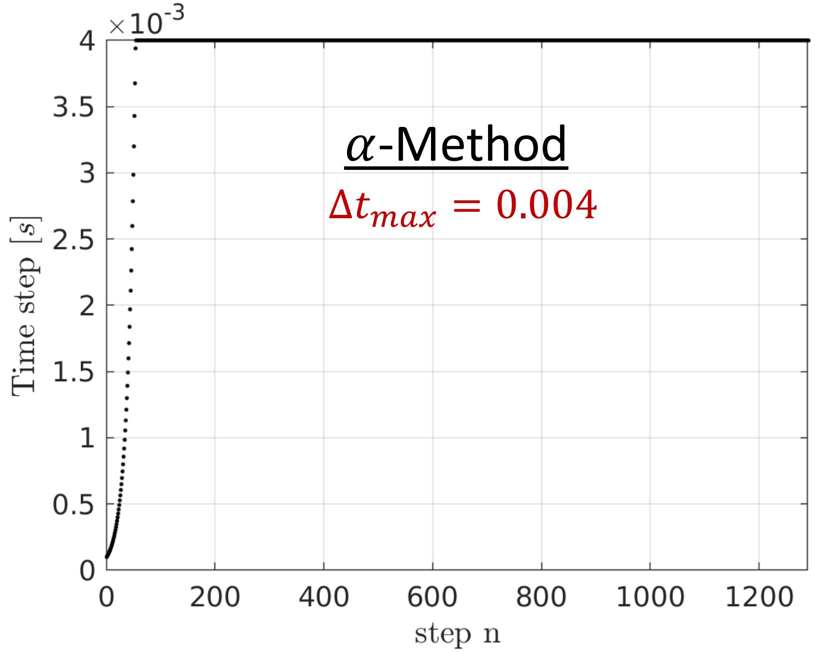


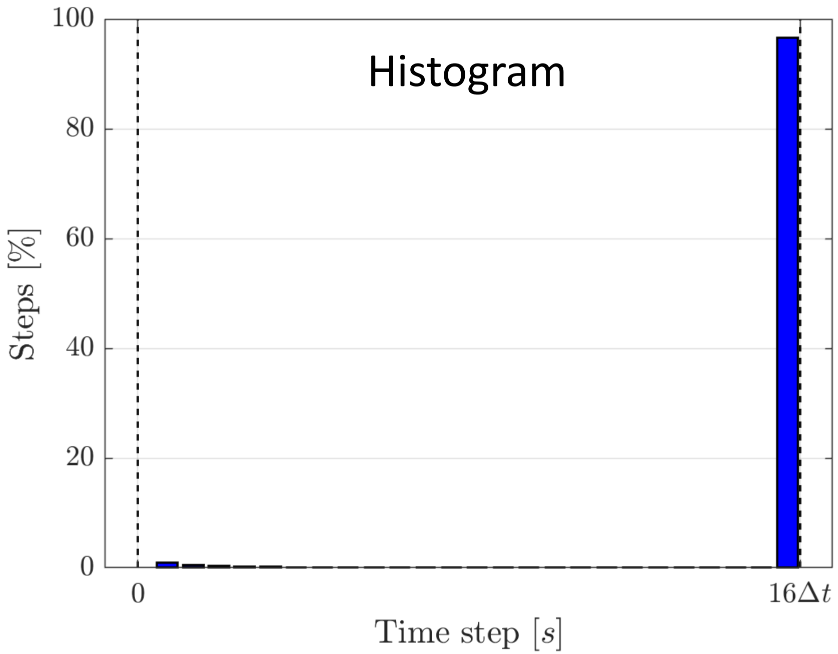


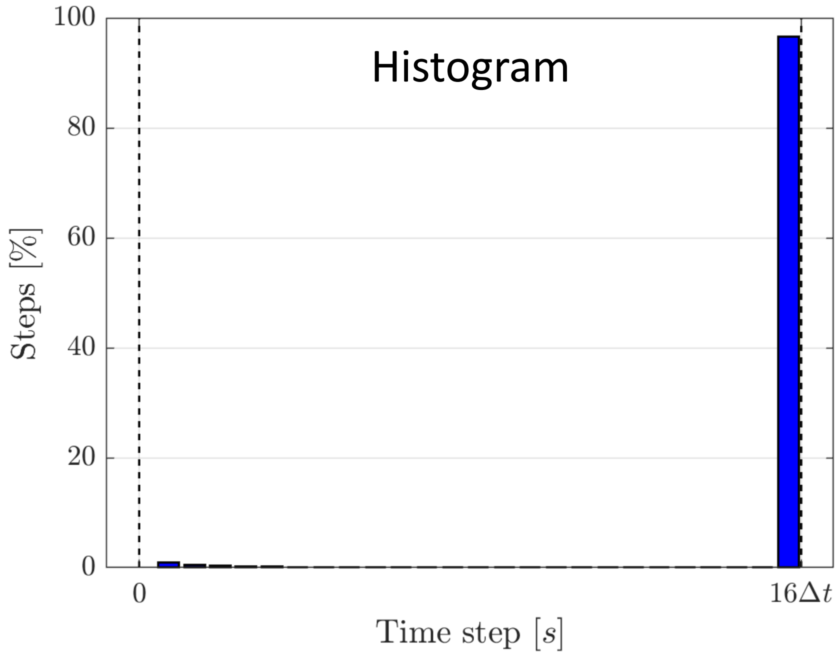


Flow Between 2 plates : Results (BE and αM)

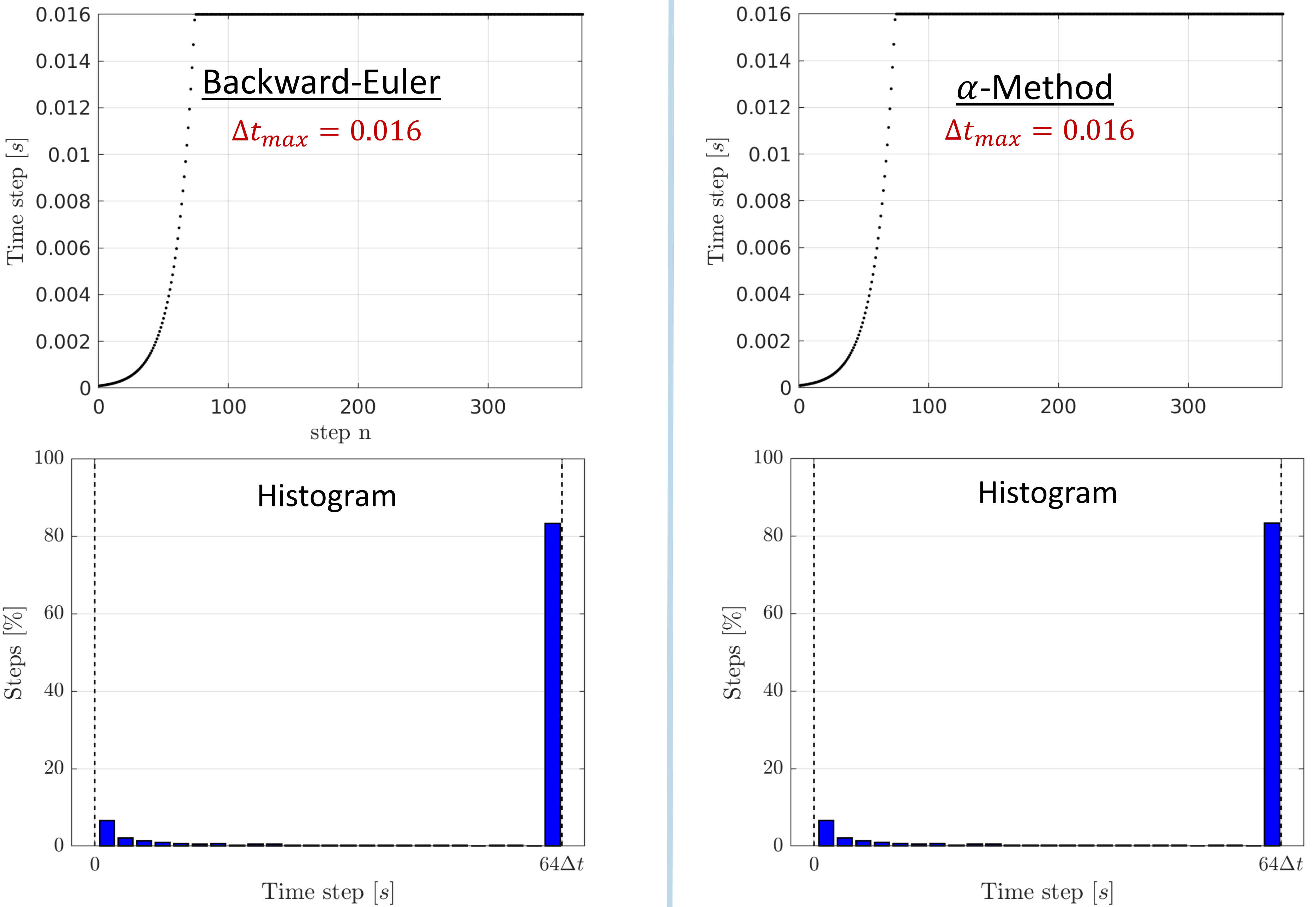




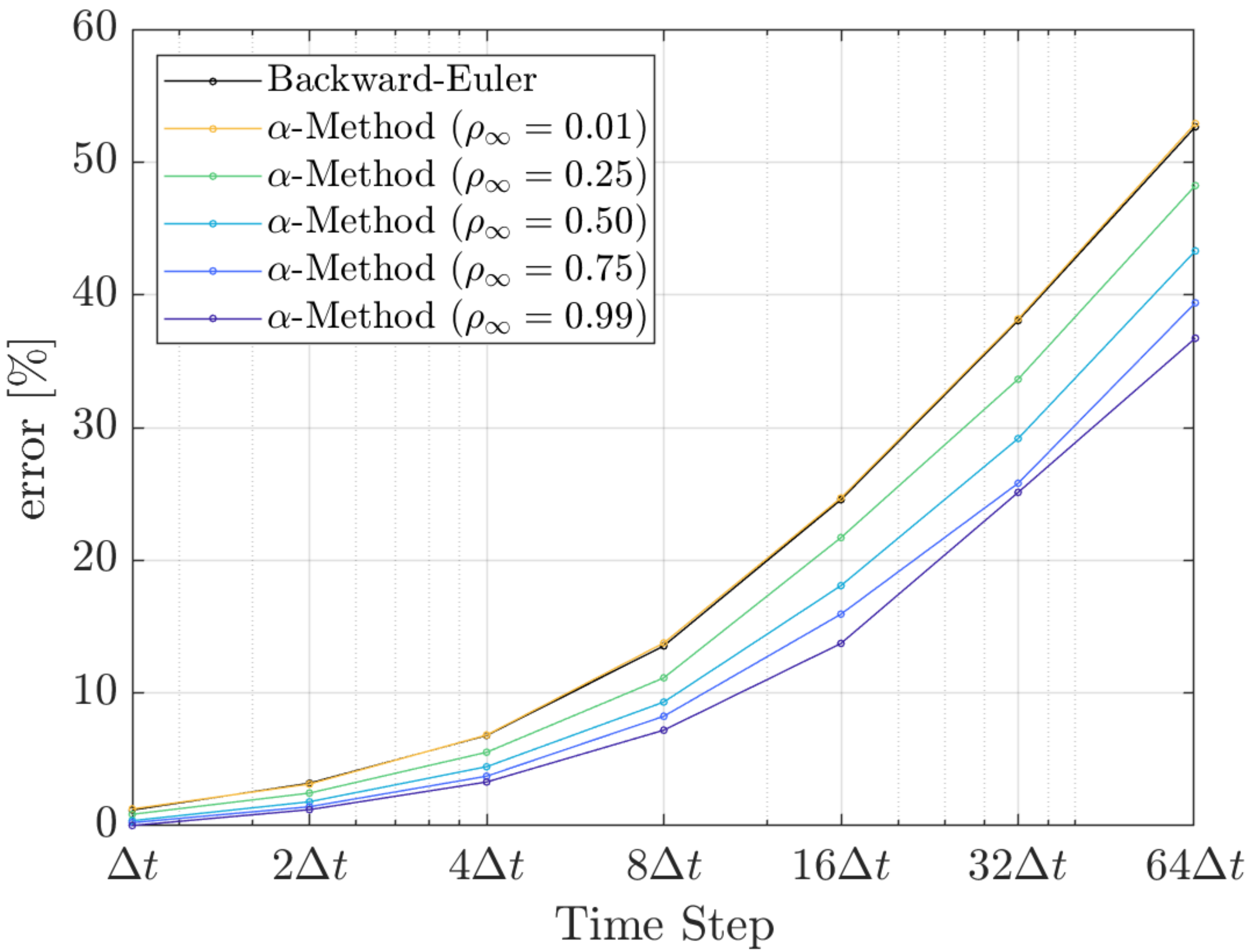




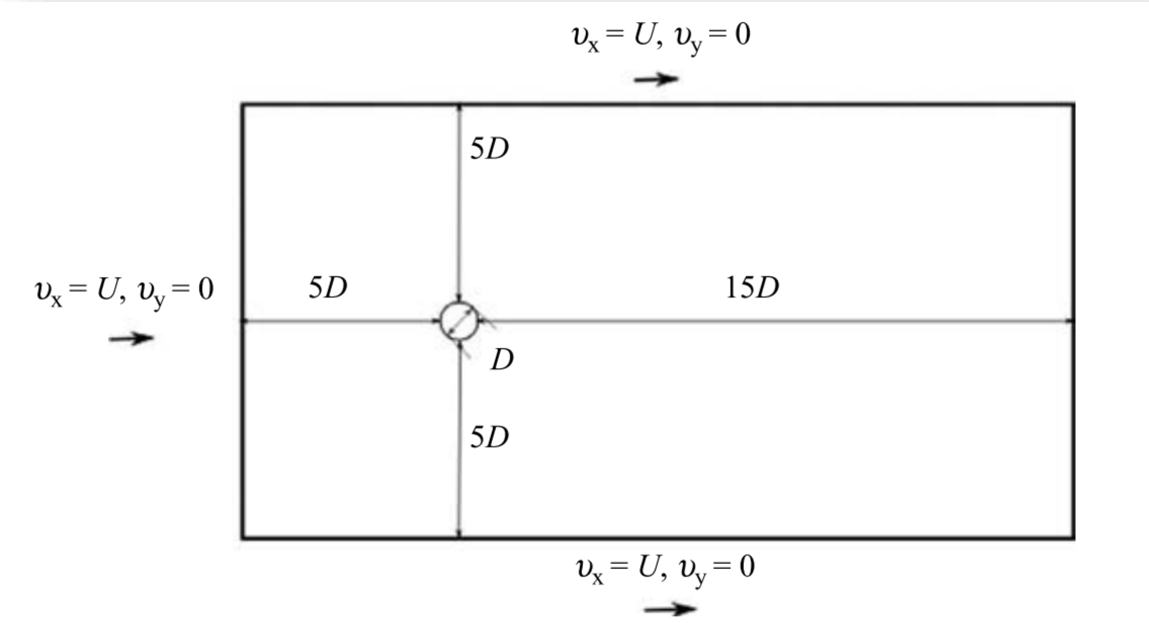
Flow Between 2 plates : Results (BE and αM)



Flow Between 2 plates : Results (BE and αM(ρ_∞))



Flow around a circular cylinder at $\mathrm{R_e} = 1000$



Idelsohn et.al. (2013)

Coefficients for comparison:

(Lift Coefficient)

$$C_L(t) = \frac{F_L(t)}{\frac{1}{2}\rho U^2 D}$$

(Drag Coefficient)

$$C_D(t) = \frac{F_D(t)}{\frac{1}{2}\rho U_\infty^2 D}$$

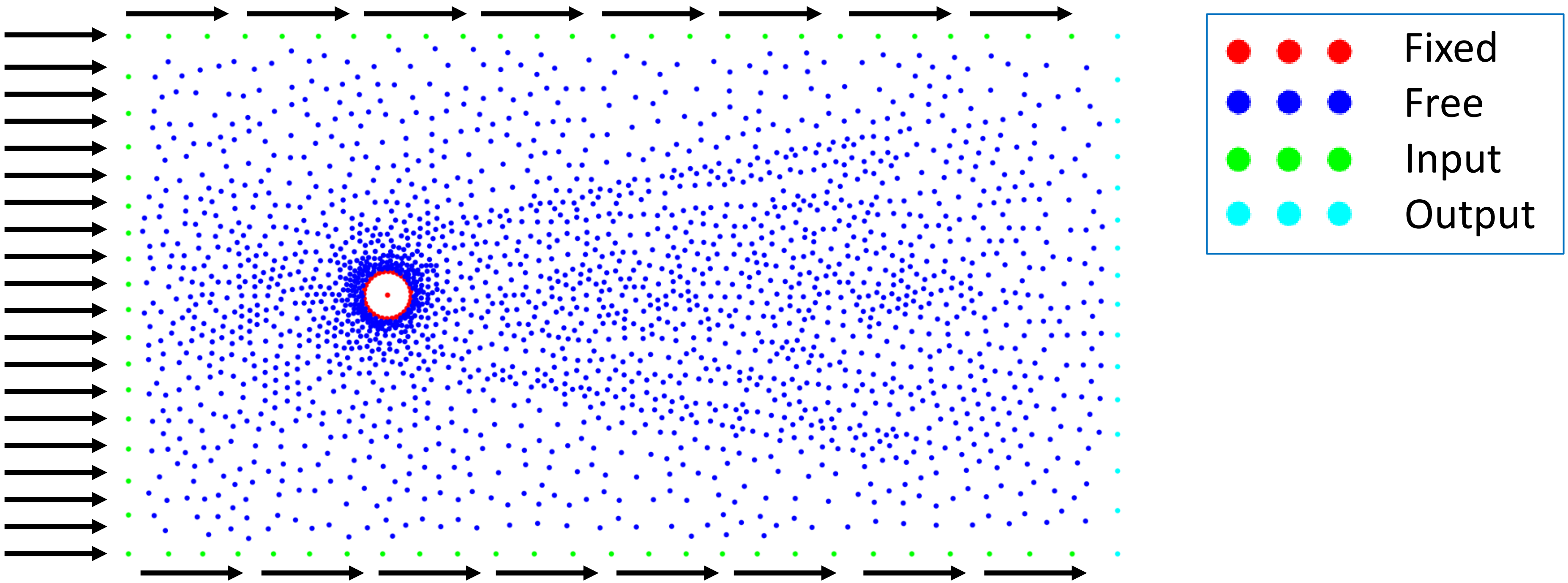
(Strouhal number)

$$S_t = \frac{fD}{U}$$

Idelsohn, S. R., Nigro, N. M., Gimenez, J. M., Rossi, R., & Marti, J. M. (2013). A fast and accurate method to solve the incompressible Navier-Stokes equations. Engineering Computations.

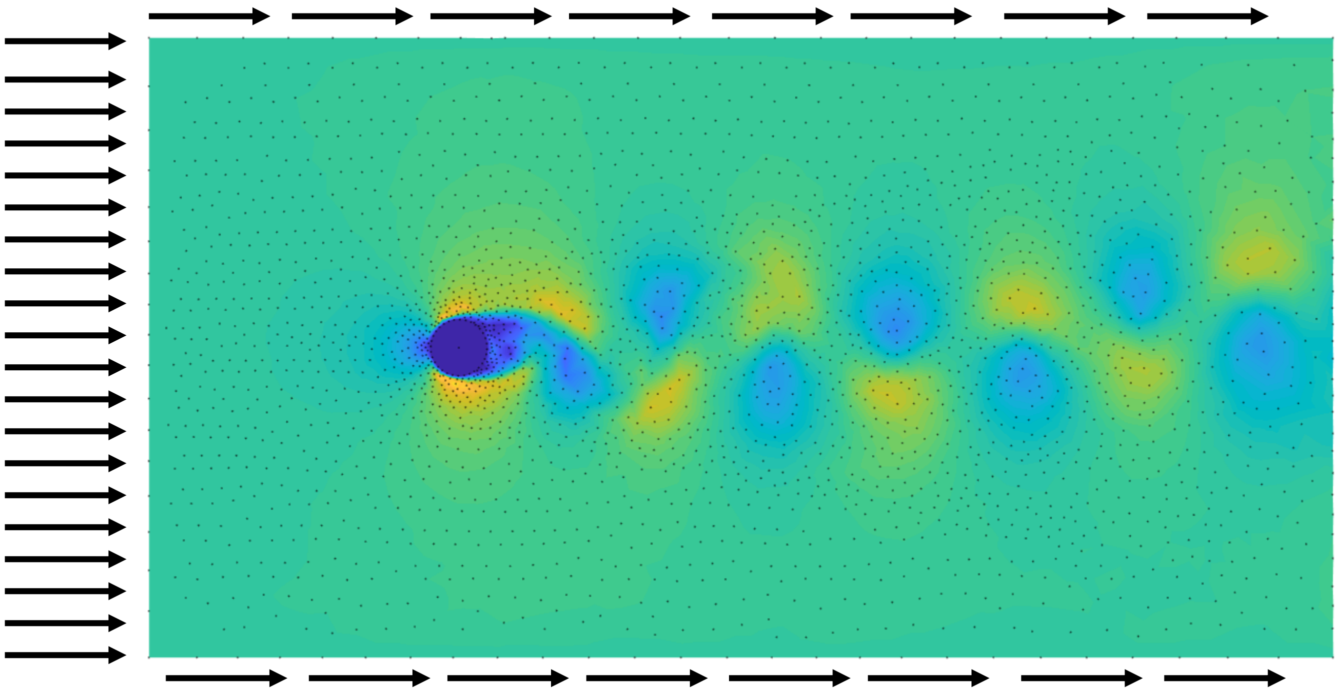
Idelsohn, S., Nigro, N., Limache, A., & Oñate, E. (2012). Large time-step explicit integration method for solving problems with dominant convection. *Computer Methods in Applied Mechanics and Engineering*, *217*, 168-185.

Flow around a circular cylinder



This work		Idelsohn et.al (2013)	Idelsohn et.al (2012)
Num. Initial Particles	: 2200	44520	77000
Input velocity	: 1 m/s	=	=
Density	: 1 kg/m^3	= (I guess)	= (I guess)
Dynamic viscosity	: 0.001 Pa s.	= (I guess)	= (I guess)
Simulation span	: 50 seconds	25 sec.	70 sec.
Maximum time step	: 0.005 / 0.01 / 0.02 / 0.04	0.025	(variable)
Initial time step	: 0.0001	PFEM-2 (Eulerian)	PFEM-2 (Lagrangian)

Flow around a circular cylinder



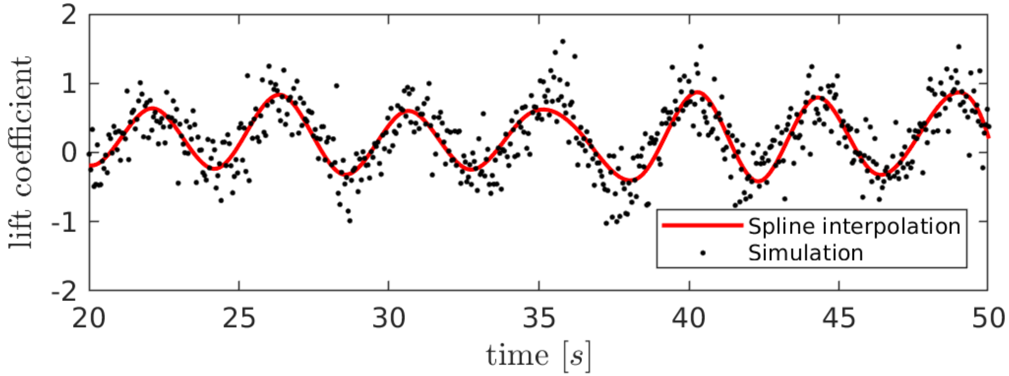
To be compared:

(Strouhal number)

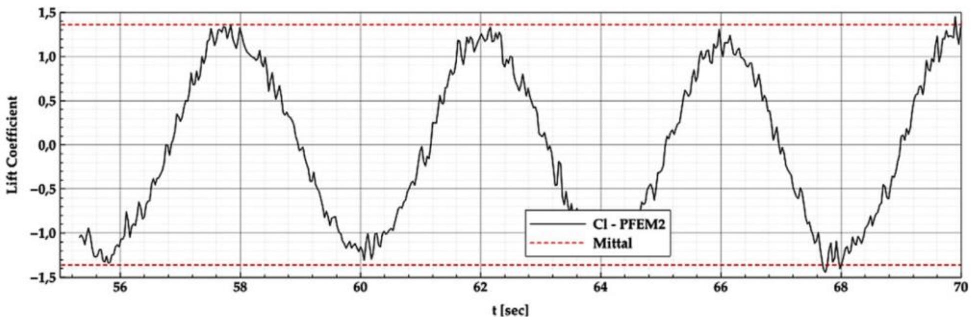
$$S_t = \frac{fD}{U}$$

This work		*Idelsohn et.al (2013)*	*Idelsohn et.al (2012)*
Num. Initial Particles	: 2200	44520	77000
Input velocity	: 1 m/s	=	=
Density	: 1 kg/m^3	= (I guess)	= (I guess)
Dynamic viscosity	: 0.001 Pa s.	= (I guess)	= (I guess)
Simulation span	: 50 seconds	25 sec.	70 sec.
Maximum time step	: 0.005 / 0.01 / 0.02 / 0.04	0.025	(variable)
Initial time step	: 0.0001	PFEM-2 (Eulerian)	PFEM-2 (Lagrangian)

Flow around a circular cylinder

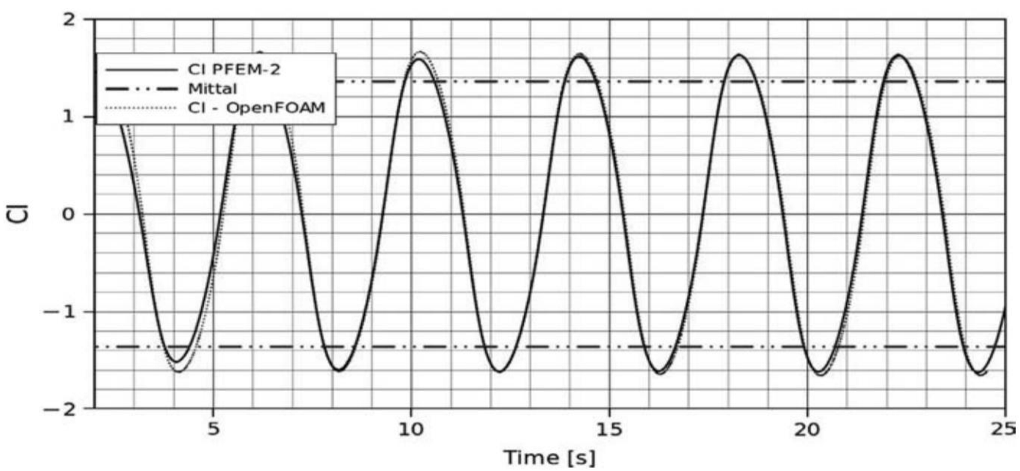


This work



Idelsohn et.al (2012) $S_t = 0.2415$

(Lagrangian)



Idelsohn et.al (2013) $S_t = 0.2475$

(Eulerian)

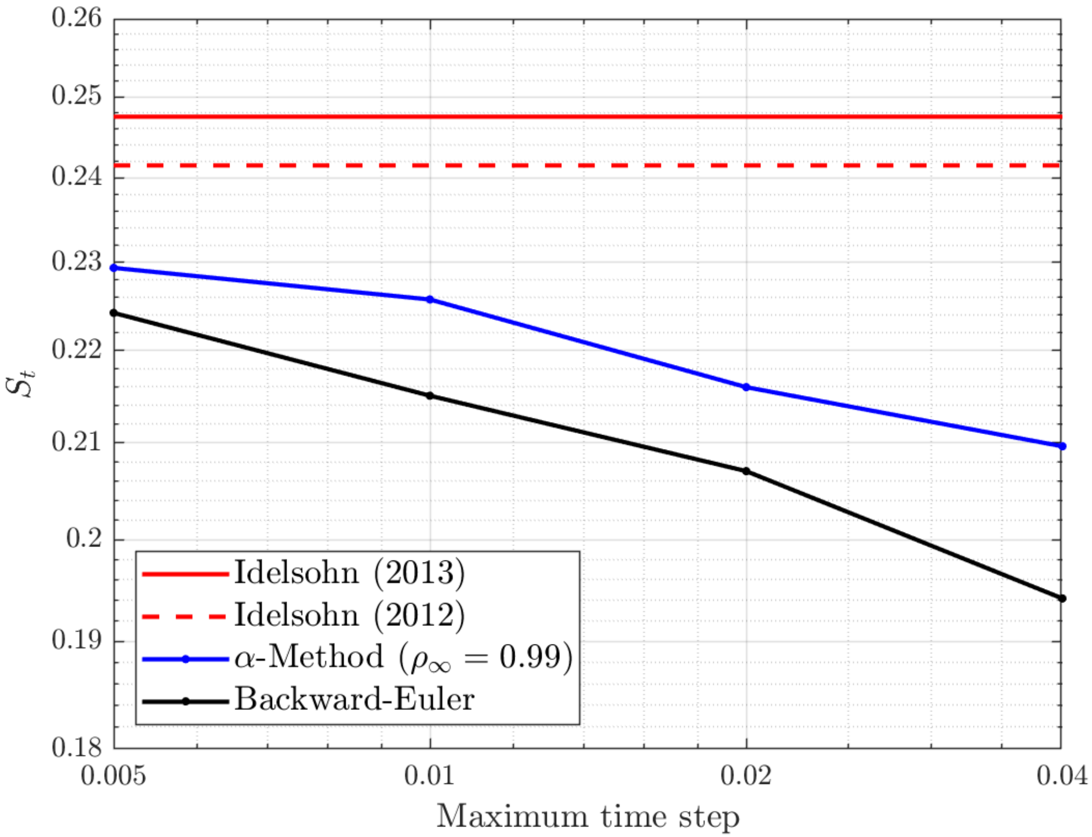
(Strouhal number)

$$S_t = \frac{fD}{U}$$

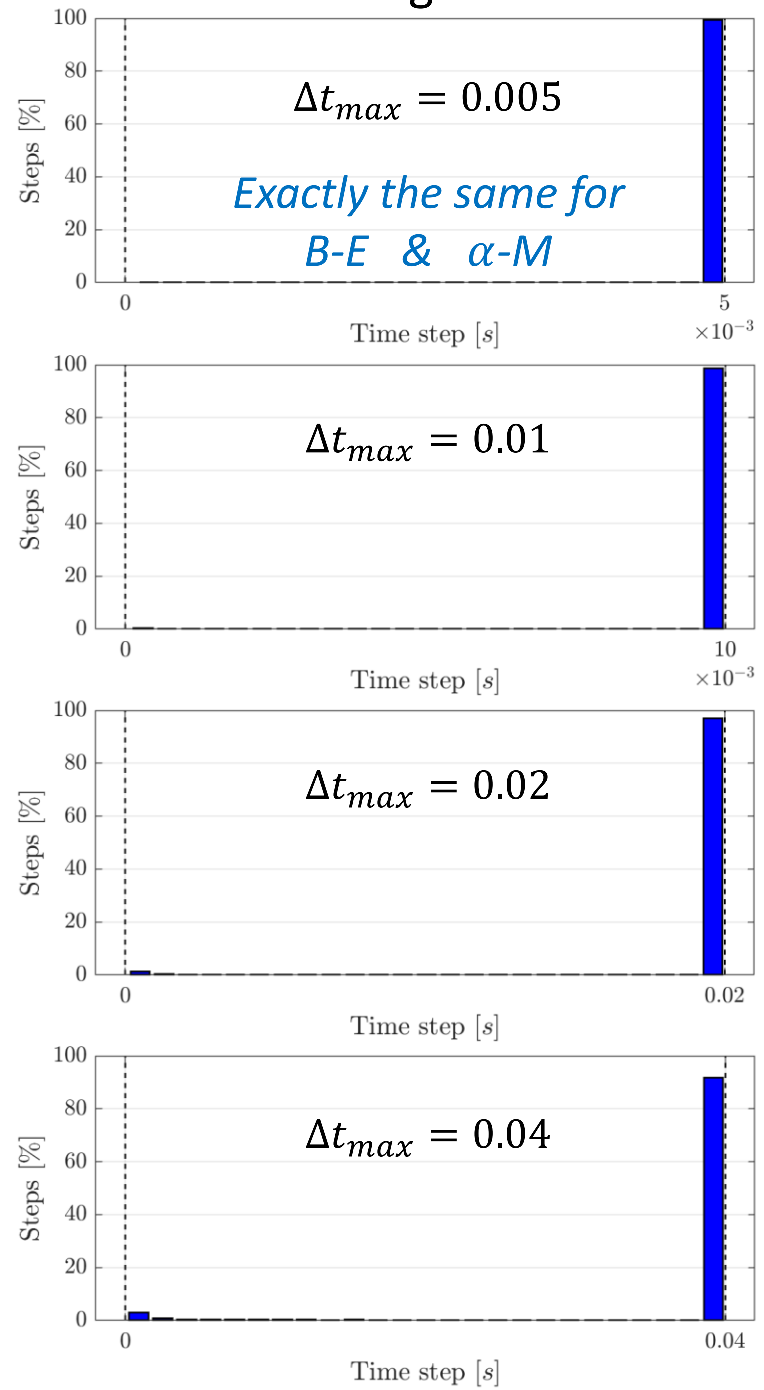
Where:

$$f = \frac{\text{cycles}}{\text{time period}}$$

Flow around a circular cylinder



Histograms



1. Time integration in PFEM

2. Results comparison: Backward-Euler & α-Method

3. Explicit time integration scheme used in PFEM-2

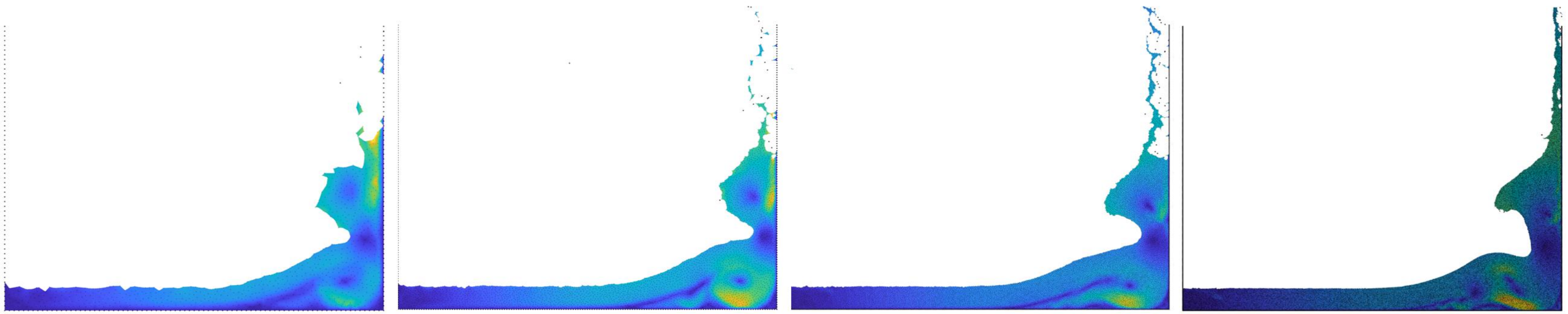
10/05/2021

Study on the explicit time integration used in PFEM-2

Motivation: Search for numerical schemes that allow the use of large time steps.

Study on the explicit time integration used in PFEM-2

Motivation: Search for numerical schemes that allow the use of large time steps.

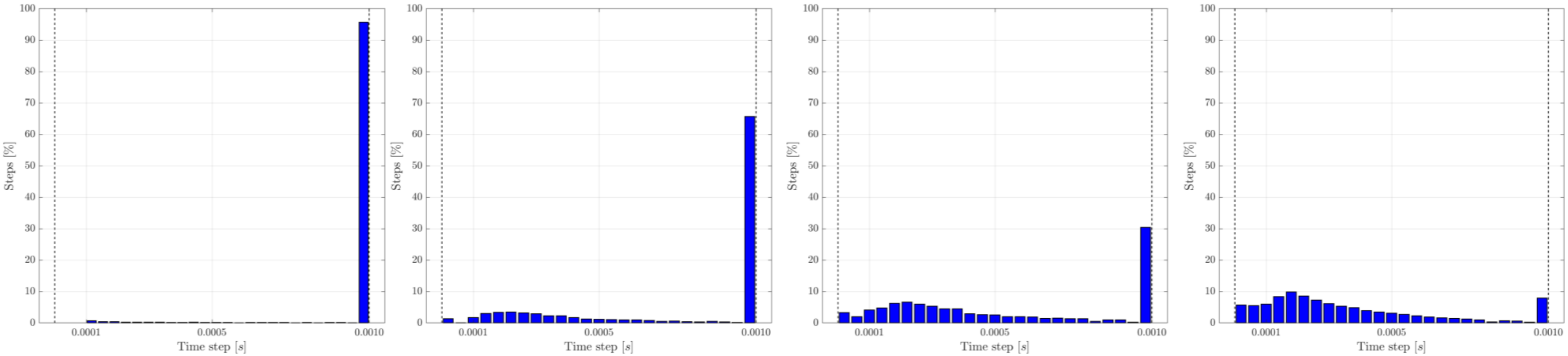


$h_{\mathrm{char}} = 0.0073$	$h_{\mathrm{char}} = 0.00365$	$h_{\mathrm{char}} = 0.001825$	$h_{\mathrm{char}} = 0.0009125$
$N_{\mathrm{part}} = 1000$	$N_{\mathrm{part}} = 3600$	$N_{\mathrm{part}} = 13600$	$N_{\mathrm{part}} = 52800$
$\Delta t_{\mathrm{max}} = 0.001$	$\Delta t_{\mathrm{max}} = 0.001$	$\Delta t_{\mathrm{max}} = 0.001$	$\Delta t_{\mathrm{max}} = 0.001$
$N_{\mathrm{steps}} = 783$	$N_{\mathrm{steps}} = 969$	$N_{\mathrm{steps}} = 1385$	$N_{\mathrm{steps}} = 2126$

(Using Backward-Euler)

Study on the explicit time integration used in PFEM-2

Motivation: Search for numerical schemes that allow the use of large time steps.

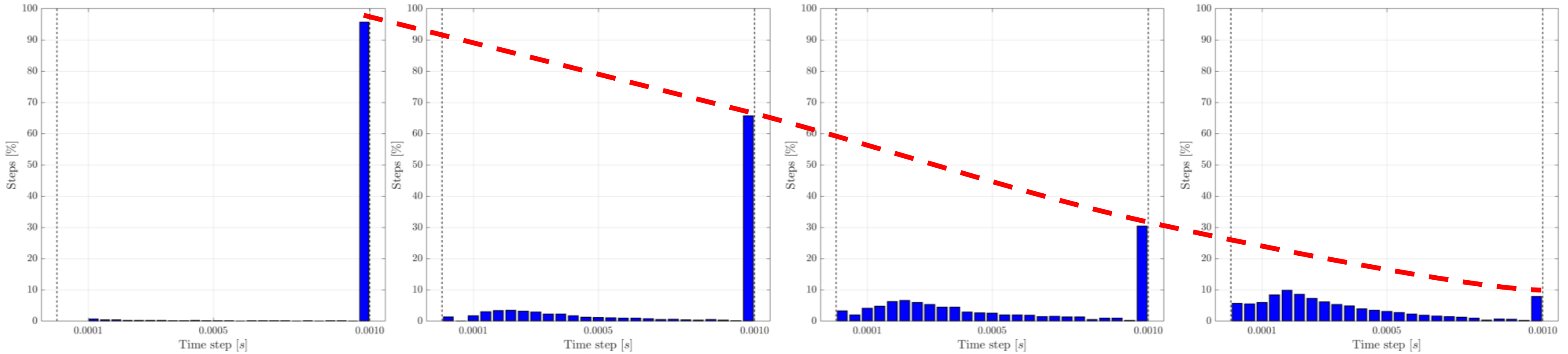


$h_{\text{char}} = 0.0073$	$h_{\text{char}} = 0.00365$	$h_{\text{char}} = 0.001825$	$h_{\text{char}} = 0.0009125$
$N_{\text{part}} = 1000$	$N_{\text{part}} = 3600$	$N_{\text{part}} = 13600$	$N_{\text{part}} = 52800$
$\Delta t_{\text{max}} = 0.001$	$\Delta t_{\text{max}} = 0.001$	$\Delta t_{\text{max}} = 0.001$	$\Delta t_{\text{max}} = 0.001$
$N_{\text{steps}} = 783$	$N_{\text{steps}} = 969$	$N_{\text{steps}} = 1385$	$N_{\text{steps}} = 2126$

(Using Backward-Euler)

Study on the explicit time integration used in PFEM-2

Motivation: Search for numerical schemes that allow the use of large time steps.



$h_{\mathrm{char}} = 0.0073$	$h_{\mathrm{char}} = 0.00365$	$h_{\mathrm{char}} = 0.001825$	$h_{\mathrm{char}} = 0.0009125$
$N_{\mathrm{part}} = 1000$	$N_{\mathrm{part}} = 3600$	$N_{\mathrm{part}} = 13600$	$N_{\mathrm{part}} = 52800$
$\Delta t_{\mathrm{max}} = 0.001$	$\Delta t_{\mathrm{max}} = 0.001$	$\Delta t_{\mathrm{max}} = 0.001$	$\Delta t_{\mathrm{max}} = 0.001$
$N_{\mathrm{steps}} = 783$	$N_{\mathrm{steps}} = 969$	$N_{\mathrm{steps}} = 1385$	$N_{\mathrm{steps}} = 2126$

(Using Backward-Euler)

Study on the explicit time integration used in PFEM-2

Known: $\mathbf{x}_{\mathrm{n}},\ \mathbf{v}_{\mathrm{n}},\ \dot{\mathbf{v}}_{\mathrm{n}}$

1) Set k = 0

2) Initial guess: $\mathbf{x}_k = \mathbf{x}_{\mathrm{n}}$

3) *While* (convergence is not reached)

3.1) Compute:

$$\mathbf{A}_{\mathrm{BE}}(\mathbf{x}_k, \mathbf{v}_k)$$

$$\mathbf{b}_{\mathrm{BE}}(\mathbf{x}_k, \mathbf{v}_{\mathrm{n}}, \mathbf{v}_{\mathrm{k}})$$

3.2) Solve : $\mathbf{A}_{\mathrm{BE}}\, \mathbf{q}_{\mathrm{n}+1} = \mathbf{b}_{\mathrm{BE}}$

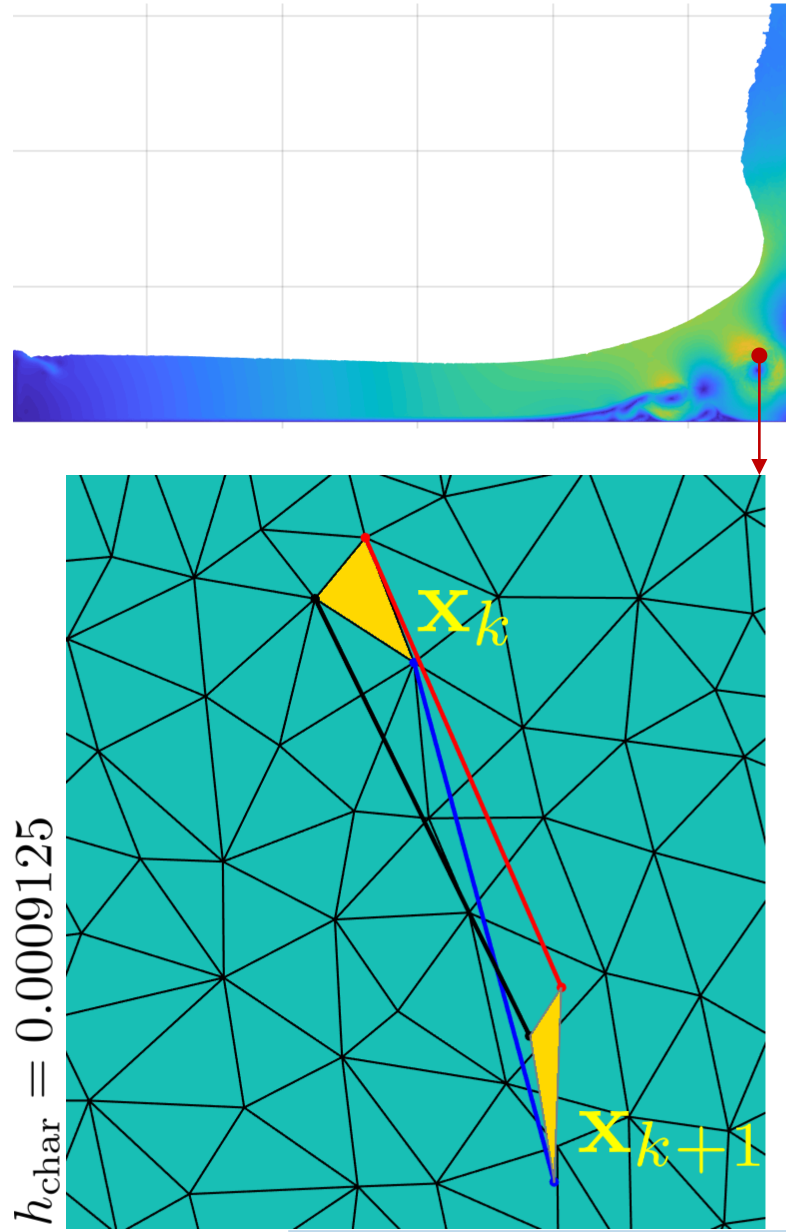
3.3) Update : $\mathbf{x}_k = \mathbf{x}_{\mathrm{n}} + \Delta t\, \mathbf{v}_{k+1}$

3.4) Set : $k = k + 1$

4) Set : $\mathbf{q}_{\mathrm{n}+1} = \mathbf{q}_{\mathrm{k}}$

Update : $\mathbf{x}_{\mathrm{n}+1} = \mathbf{x}_{\mathrm{n}} + \Delta t\, \mathbf{v}_{\mathrm{n}+1}$

5) RE-MESH (if necessary)



Study on the explicit time integration used in PFEM-2

Known: $\mathbf{x}_\mathrm{n},\ \mathbf{v}_\mathrm{n},\ \dot{\mathbf{v}}_\mathrm{n}$

1) Set k = 0

2) Initial guess: $\mathbf{x}_k = \mathbf{x}_\mathrm{n}$

3) *While* (convergence is not reached)

   3.1) Compute:

   $$\mathbf{A}_\mathrm{BE}(\mathbf{x}_k, \mathbf{v}_k)$$

   $$\mathbf{b}_\mathrm{BE}(\mathbf{x}_k, \mathbf{v}_\mathrm{n}, \mathbf{v}_\mathrm{k})$$

   3.2) Solve : $\mathbf{A}_\mathrm{BE}\, \mathbf{q}_\mathrm{n+1} = \mathbf{b}_\mathrm{BE}$

   3.3) Update : $\mathbf{x}_k = \mathbf{x}_\mathrm{n} + \Delta t\, \mathbf{v}_{k+1}$

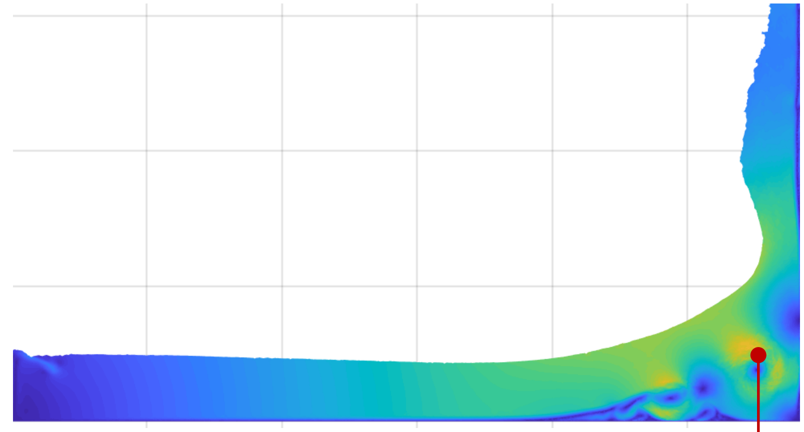
   + RE-MESH (if necessary)

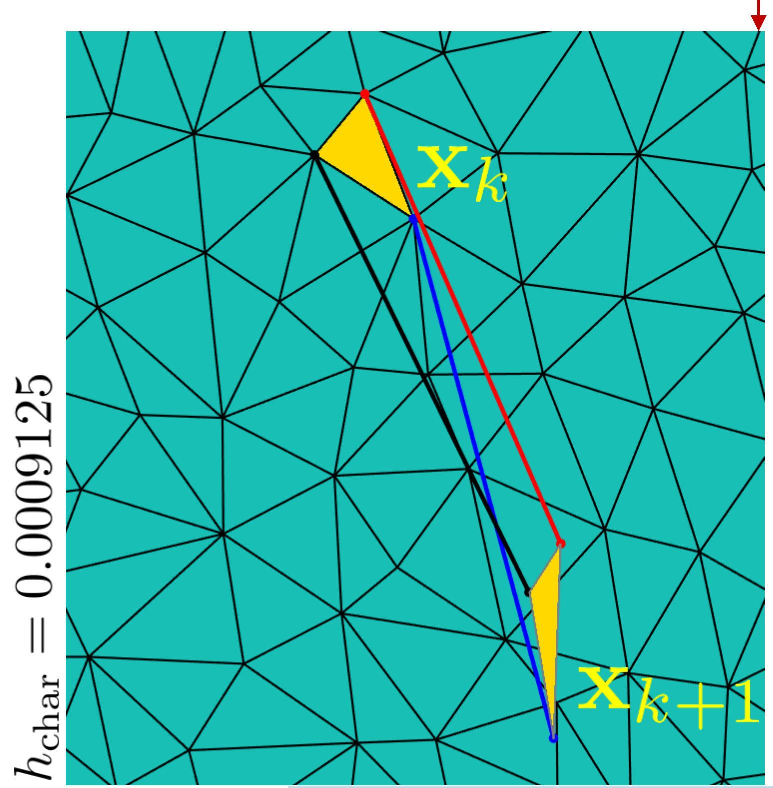
   3.4) Set : $k = k + 1$

4) Set : $\mathbf{q}_\mathrm{n+1} = \mathbf{q}_\mathrm{k}$

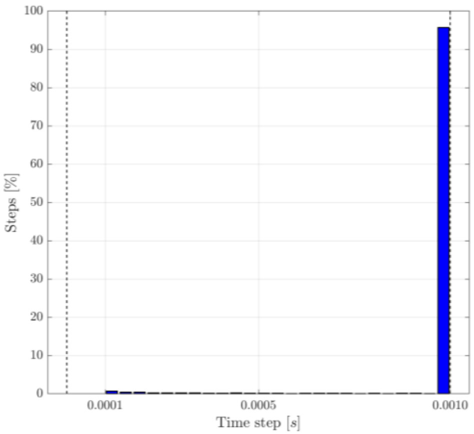
   Update : $\mathbf{x}_\mathrm{n+1} = \mathbf{x}_\mathrm{n} + \Delta t\, \mathbf{v}_\mathrm{n+1}$

5) RE-MESH (if necessary)



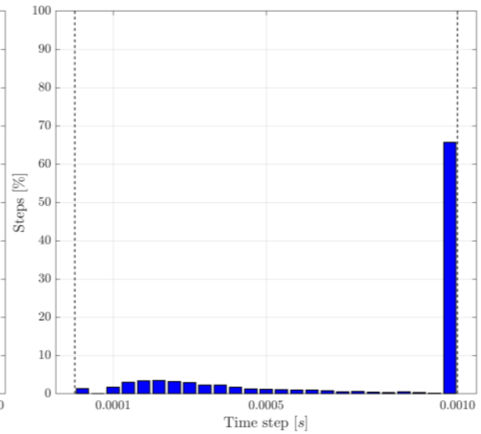


Study on the explicit time integration used in PFEM-2



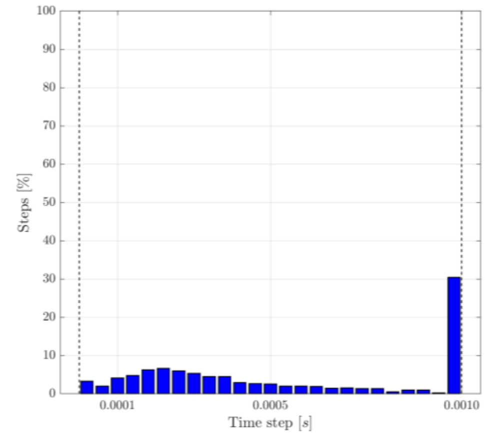
$h_{\text{char}} = 0.0073$

$N_{\text{steps}} = 783$



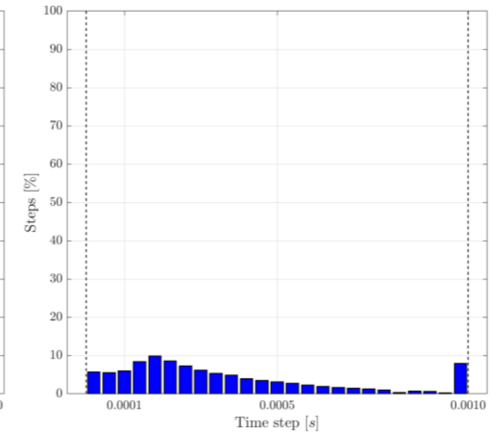
$h_{\text{char}} = 0.00365$

$N_{\text{steps}} = 969$



$h_{\text{char}} = 0.001825$

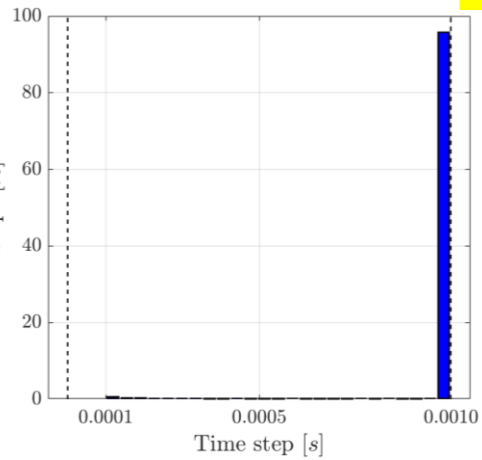
$N_{\text{steps}} = 1385$



$h_{\text{char}} = 0.0009125$

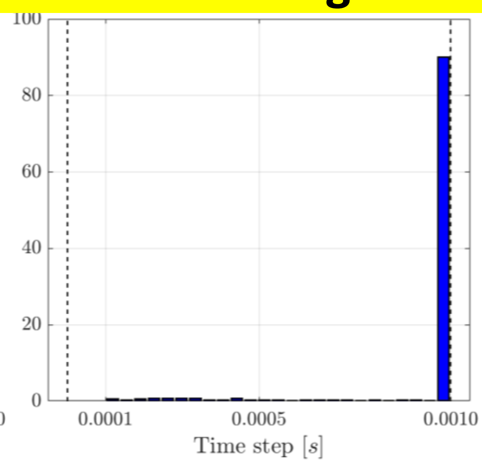
$N_{\text{steps}} = 2126$

With Remeshing inside non-linear algorithm



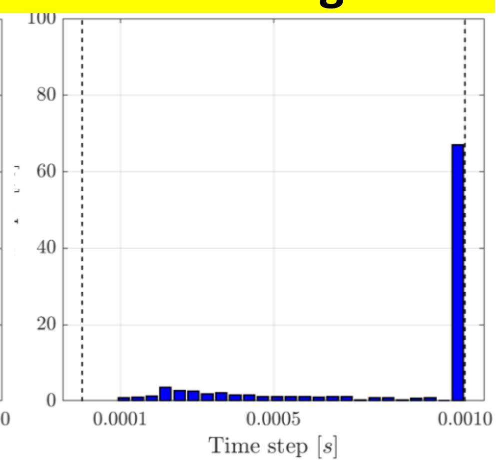
$h_{\text{char}} = 0.0073$

$N_{\text{steps}} = 771$



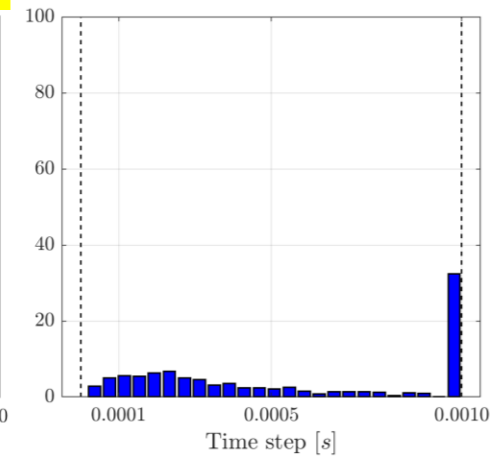
$h_{\text{char}} = 0.00365$

$N_{\text{steps}} = 798$



$h_{\text{char}} = 0.001825$

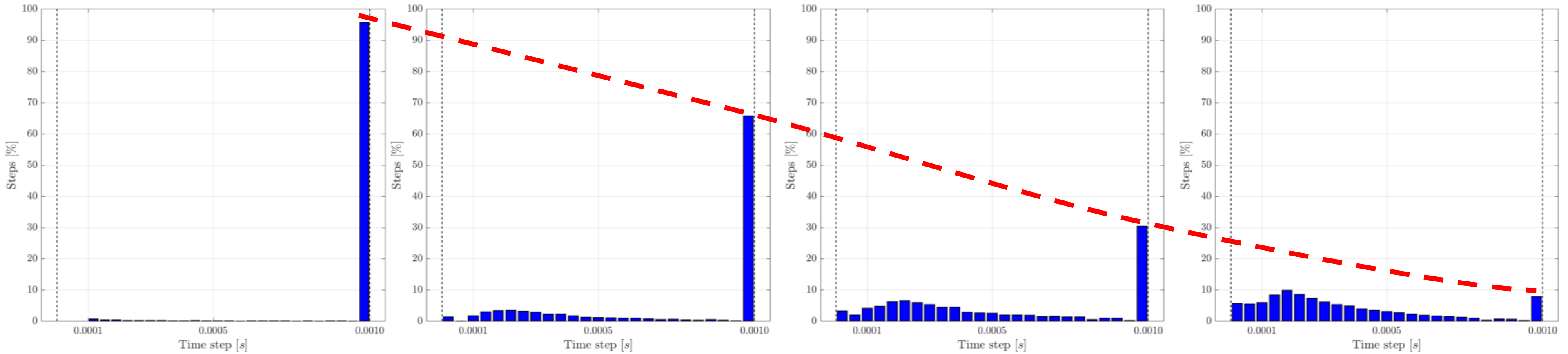
$N_{\text{steps}} = 943$



$h_{\text{char}} = 0.0009125$

$N_{\text{steps}} = 1442$

Study on the explicit time integration used in PFEM-2



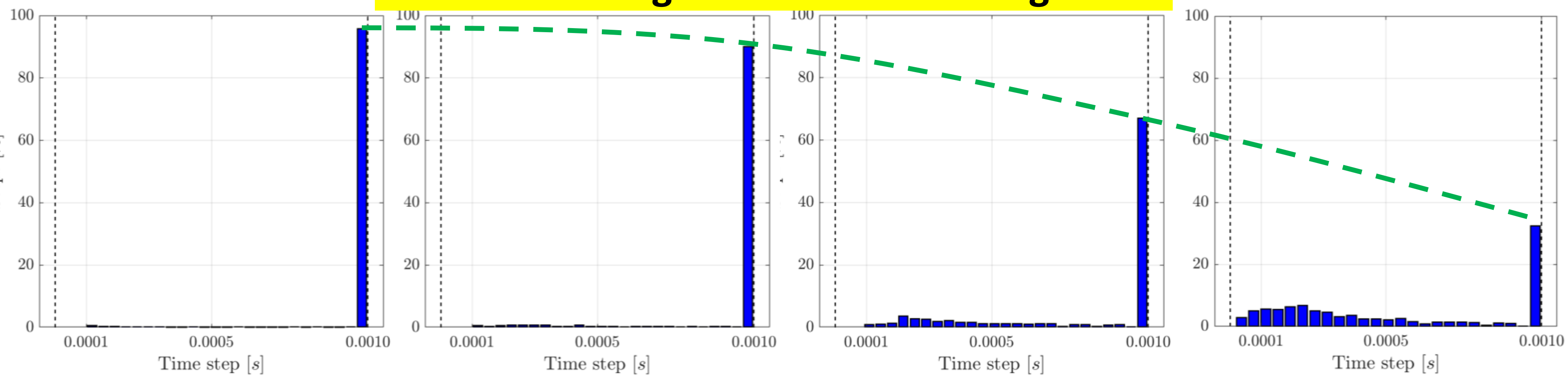
$h_{\text{char}} = 0.0073$ $N_{\text{steps}} = 783$

$h_{\text{char}} = 0.00365$ $N_{\text{steps}} = 969$

$h_{\text{char}} = 0.001825$ $N_{\text{steps}} = 1385$

$h_{\text{char}} = 0.0009125$ $N_{\text{steps}} = 2126$

With Remeshing inside non-linear algorithm



$h_{\text{char}} = 0.0073$ $N_{\text{steps}} = 771$

$h_{\text{char}} = 0.00365$ $N_{\text{steps}} = 798$

$h_{\text{char}} = 0.001825$ $N_{\text{steps}} = 943$

$h_{\text{char}} = 0.0009125$ $N_{\text{steps}} = 1442$

Study on the explicit time integration used in PFEM-2

Known: $\mathbf{x}_{\mathrm{n}},\ \mathbf{v}_{\mathrm{n}},\ \dot{\mathbf{v}}_{\mathrm{n}}$

1) Set k = 0

2) Initial guess: $\mathbf{x}_k = \mathbf{x}_{\mathrm{n}}$

3) *While* (convergence is not reached)

3.1) Compute:

$$\mathbf{A}_{\mathrm{BE}}(\mathbf{x}_k, \mathbf{v}_k)$$

$$\mathbf{b}_{\mathrm{BE}}(\mathbf{x}_k, \mathbf{v}_{\mathrm{n}}, \mathbf{v}_{\mathrm{k}})$$

3.2) Solve : $\mathbf{A}_{\mathrm{BE}}\,\mathbf{q}_{\mathrm{n}+1} = \mathbf{b}_{\mathrm{BE}}$

3.3) Update : $\mathbf{x}_k = \mathbf{x}_{\mathrm{n}} + \Delta t\, \mathbf{v}_{k+1}$

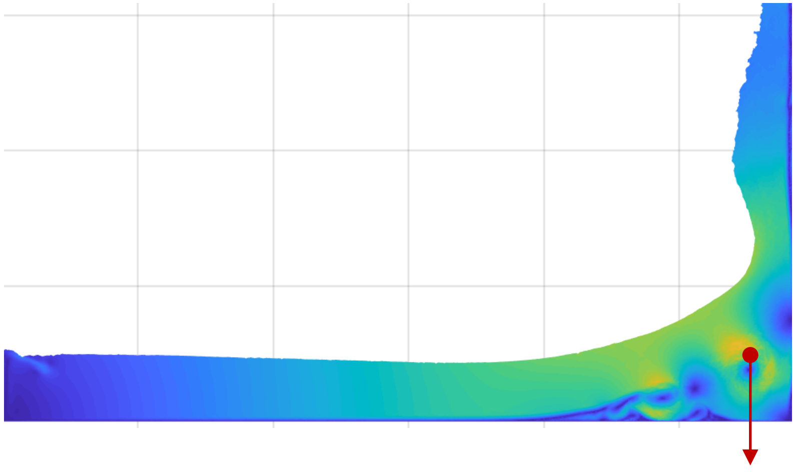
+ RE-MESH (if necessary)

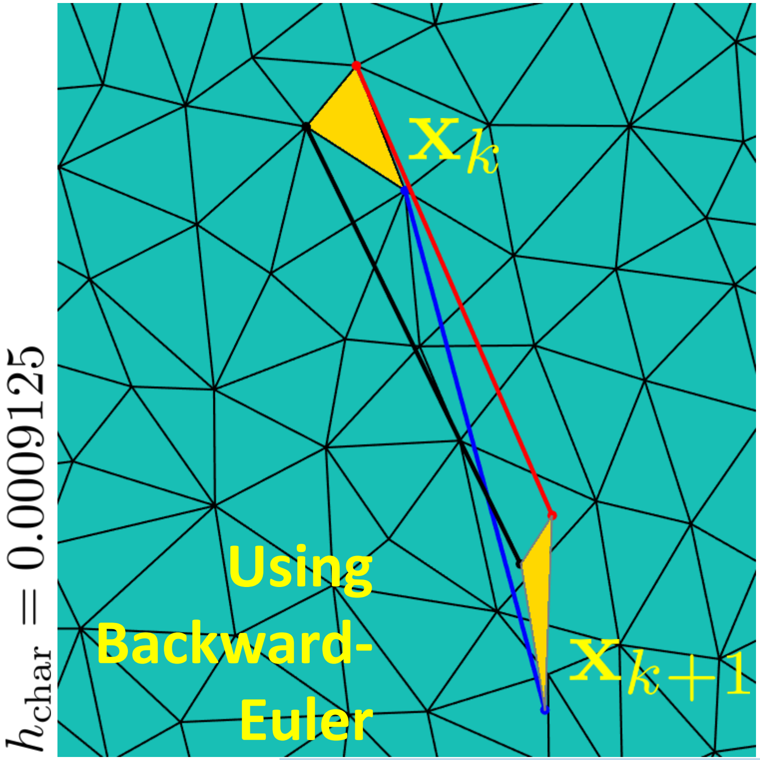
3.4) Set : $k = k + 1$

4) Set : $\mathbf{q}_{\mathrm{n}+1} = \mathbf{q}_{\mathrm{k}}$

Update : $\mathbf{x}_{\mathrm{n}+1} = \mathbf{x}_{\mathrm{n}} + \Delta t\, \mathbf{v}_{\mathrm{n}+1}$

5) RE-MESH (if necessary)





Study on the explicit time integration used in PFEM-2

Known: $\mathbf{x}_{\mathrm{n}},\ \mathbf{v}_{\mathrm{n}},\ \dot{\mathbf{v}}_{\mathrm{n}}$

1) Set k = 0

2) Initial guess: $\mathbf{x}_k = \mathbf{x}_{\mathrm{n}}$

3) *While* (convergence is not reached)

3.1) Compute:

$$\mathbf{A}_{\mathrm{BE}}(\mathbf{x}_k, \mathbf{v}_k)$$

$$\mathbf{b}_{\mathrm{BE}}(\mathbf{x}_k, \mathbf{v}_{\mathrm{n}}, \mathbf{v}_{\mathrm{k}})$$

3.2) Solve : $\mathbf{A}_{\mathrm{BE}}\, \mathbf{q}_{\mathrm{n+1}} = \mathbf{b}_{\mathrm{BE}}$

3.3) Update : $\mathbf{x}_k$ → Use a more accurate Integration scheme

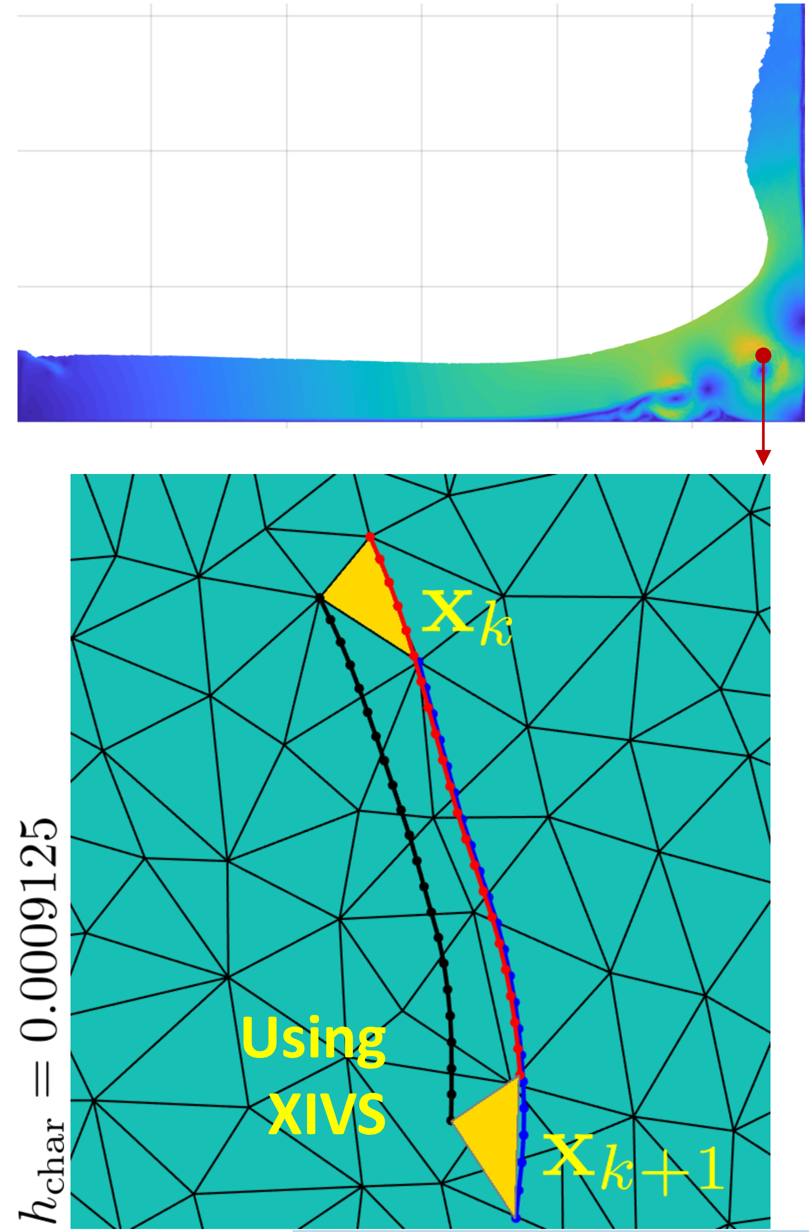
+ RE-MESH (if necessary)

3.4) Set : $k = k + 1$

4) Set : $\mathbf{q}_{\mathrm{n+1}} = \mathbf{q}_{\mathrm{k}}$

Update : $\mathbf{x}_{\mathrm{n+1}} = \mathbf{x}_{\mathrm{n}} + \Delta t\, \mathbf{v}_{\mathrm{n+1}}$

5) RE-MESH (if necessary)



Study on the explicit time integration used in PFEM-2

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau \quad (1)$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{n} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau \quad (2)$$

2) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ ~~and around~~ $\cancel{\mathbf{x}^{\mathrm{p}}}$

See, for instance:

Dialami, N., Chiumenti, M., Cervera, M., De Saracibar, C. A., & Ponthot, J. P. (2015). Material flow visualization in friction stir welding via particle tracing. *International Journal of Material Forming*, *8*(2), 167-181.

Marti, J., & Ryzhakov, P. (2020). Improving accuracy of the moving grid particle finite element method via a scheme based on Strang splitting. *Computer Methods in Applied Mechanics and Engineering*, *369*, 113212.

Idelsohn, S., Nigro, N., Limache, A., & Oñate, E. (2012). Large time-step explicit integration method for solving problems with dominant convection. *Computer Methods in Applied Mechanics and Engineering*, *217*, 168-185.

Study on the explicit time integration used in PFEM-2

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau \qquad (1)$$

$$\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}} = \mathbf{x}^{\mathrm{p}}_{n} + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \mathbf{v}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau \qquad (2)$$

2) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau})$ constant between $[t_{\mathrm{n}}, t_{\mathrm{n+1}}]$ ~~and around~~ $\cancel{\mathbf{x}^{\mathrm{p}}}$

2.a) Assume $\dot{\mathbf{v}}(\tau, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}(t_{\mathrm{n}}, \mathbf{x}^{\mathrm{p}}_{\tau}) = \dot{\mathbf{v}}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\tau})$

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} \dot{\mathbf{v}}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{\tau})\mathrm{d}\tau \qquad (3)$$

Space Discretization:

$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \int_{t_{\mathrm{n}}}^{t_{\mathrm{n+1}}} N_I(\mathbf{x}^{\mathrm{p}}_{\tau})\dot{\mathbf{v}}_{\mathrm{n},I}\mathrm{d}\tau \qquad (4)$$

Study on the explicit time integration used in PFEM-2

Time Discretization:

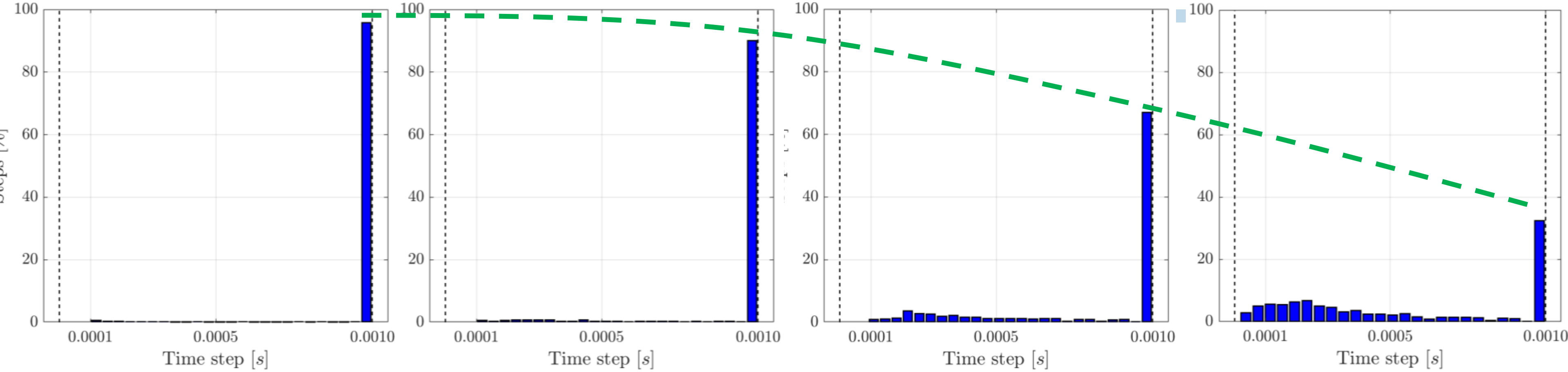
$$\mathbf{v}_{\mathrm{n+1}}(\mathbf{x}^{\mathrm{p}}_{\mathrm{n+1}}) = \mathbf{v}_{\mathrm{n}}(\mathbf{x}^{\mathrm{p}}_{n}) + \sum_{i=1}^{m} \delta t N_I(\mathbf{x}^{\mathrm{p}}_{\mathrm{n}+\frac{i}{m}})\dot{\mathbf{v}}_{\mathrm{n},I} \qquad (5)$$

Where :

m is the number of sub-steps discretizing the time step Δt

δt is the sub-time step computed as : $\delta t = \dfrac{\Delta t}{m}$

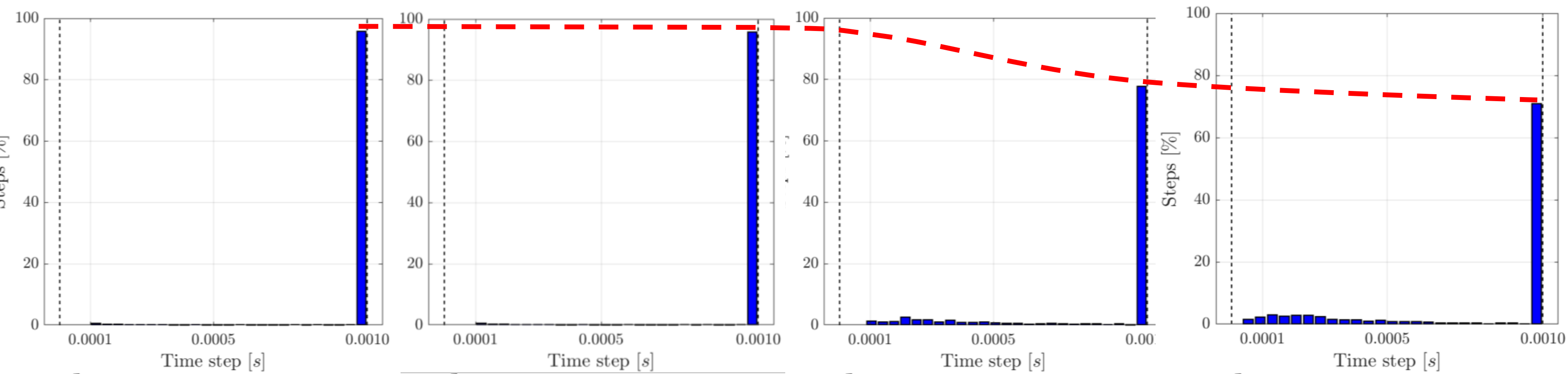
With Remeshing inside non-linear algorithm



$h_{\text{char}} = 0.0073$ | $h_{\text{char}} = 0.00365$ | $h_{\text{char}} = 0.001825$ | $h_{\text{char}} = 0.0009125$

$N_{\text{steps}} = 771$ | $N_{\text{steps}} = 798$ | $N_{\text{steps}} = 943$ | $N_{\text{steps}} = 1442$

With Remeshing inside non-linear algorithm + more accurate Integration (for position)

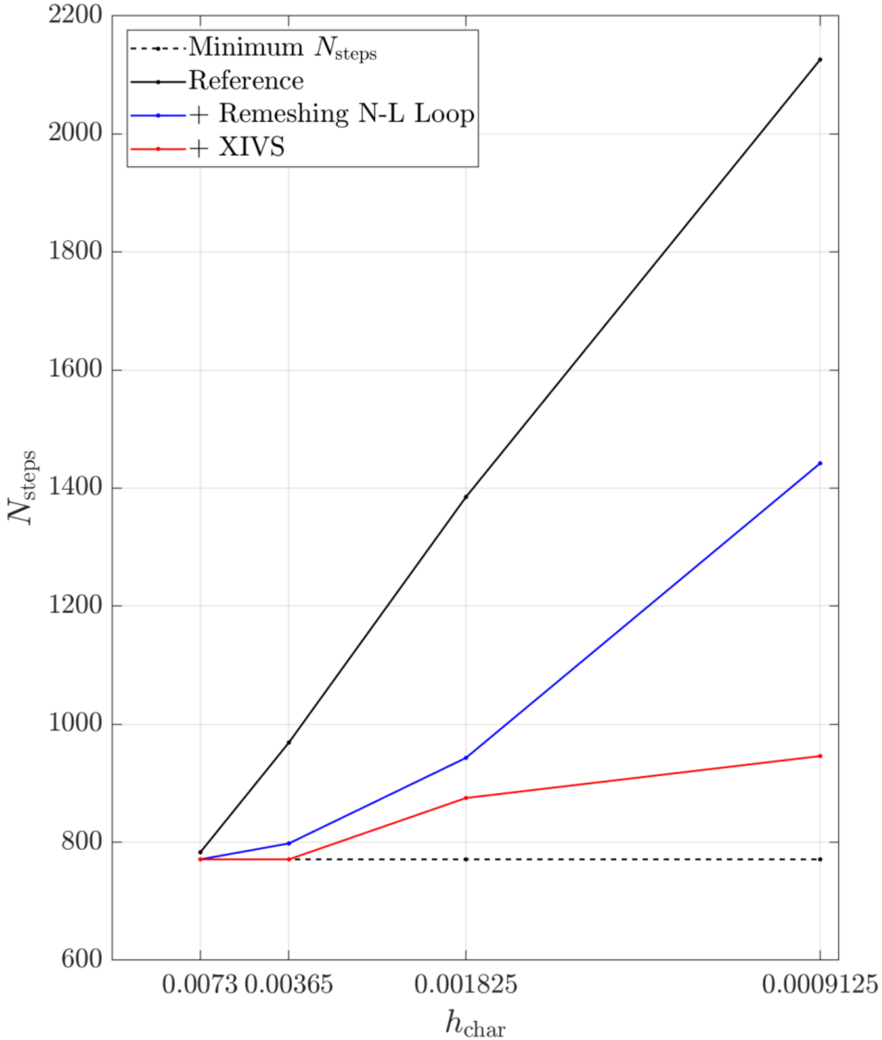


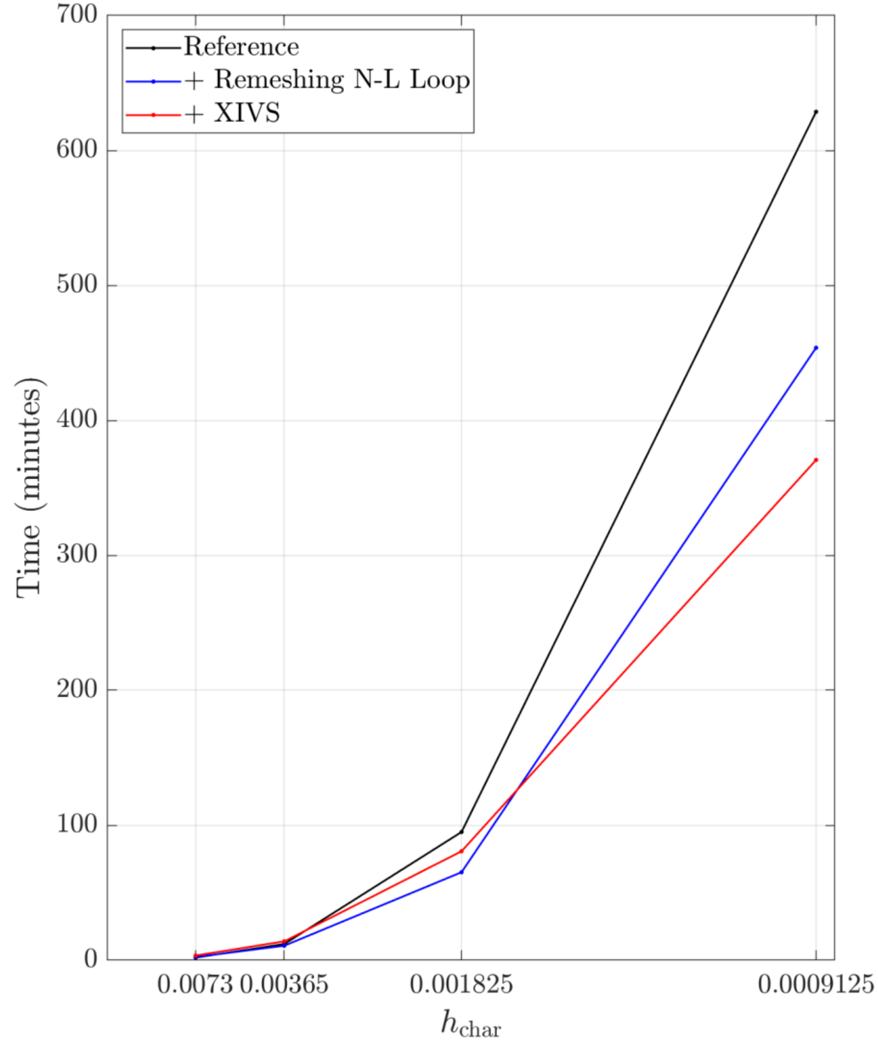
$h_{\text{char}} = 0.0073$ | $h_{\text{char}} = 0.00365$ | $h_{\text{char}} = 0.001825$ | $h_{\text{char}} = 0.0009125$

$N_{\text{steps}} = 771$ | $N_{\text{steps}} = 771$ | $N_{\text{steps}} = 875$ | $N_{\text{steps}} = 946$

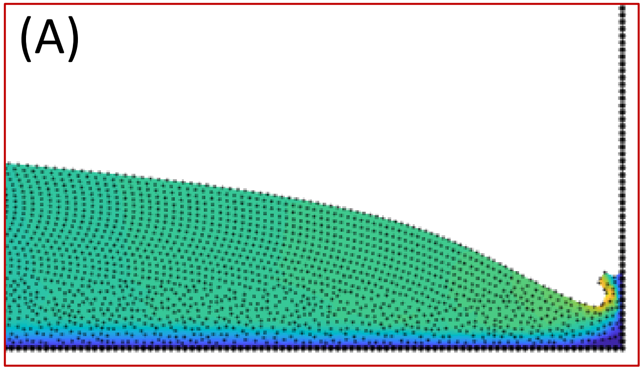
Study on the explicit time integration used in PFEM-2

From Dam Break:



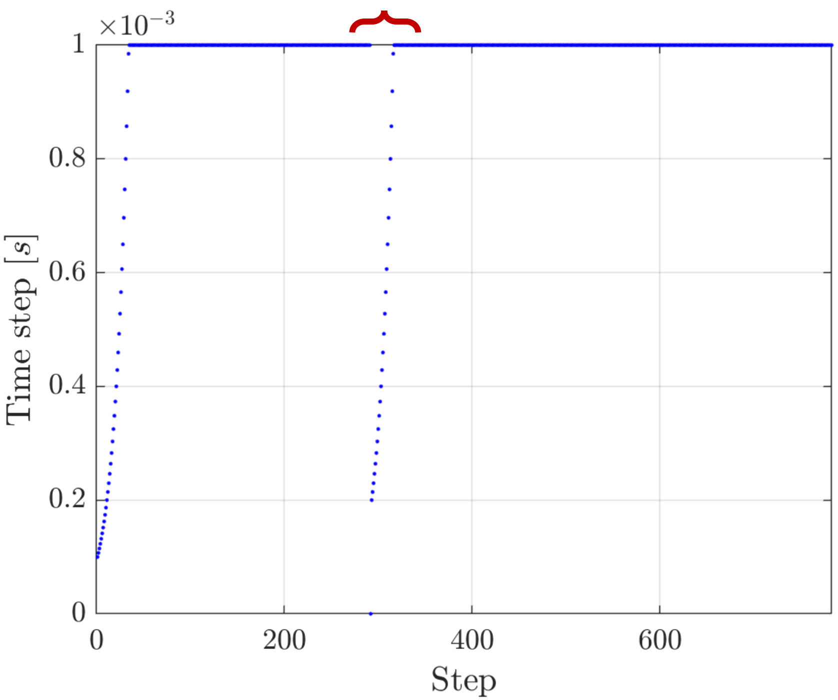


Study on the explicit time integration used in PFEM-2

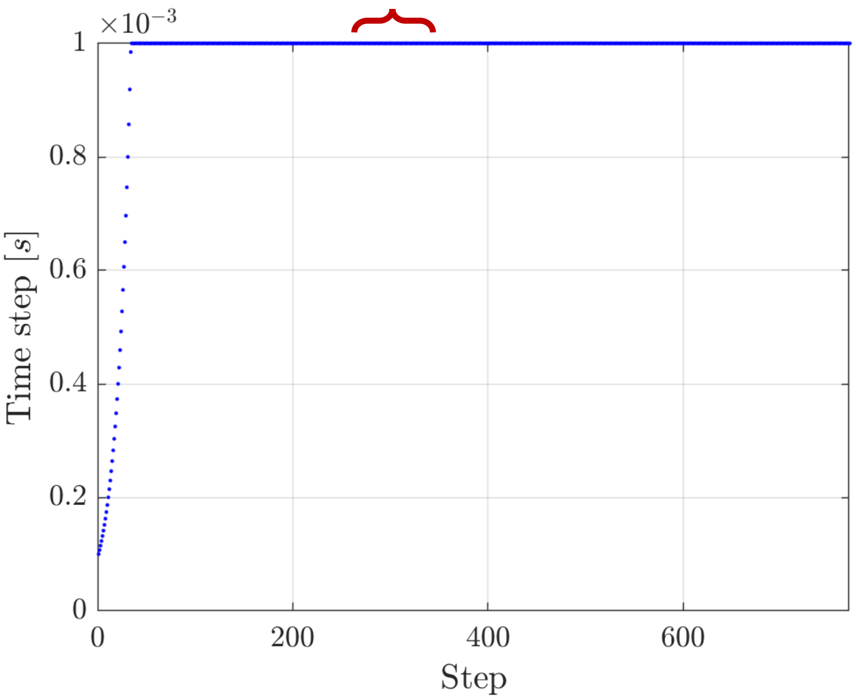


Reference $(h_{\mathrm{char}} = 0.0073)$ **+ Remeshing (N-L loop) + XIVS integration**

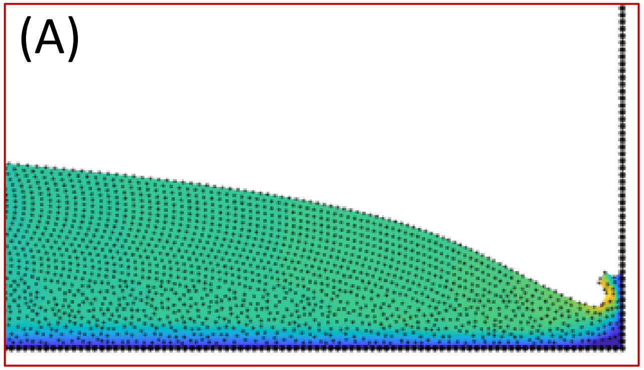
(A)

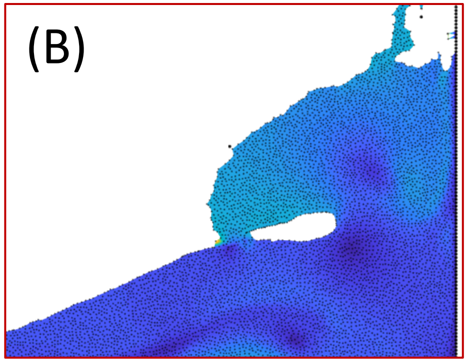


(A)

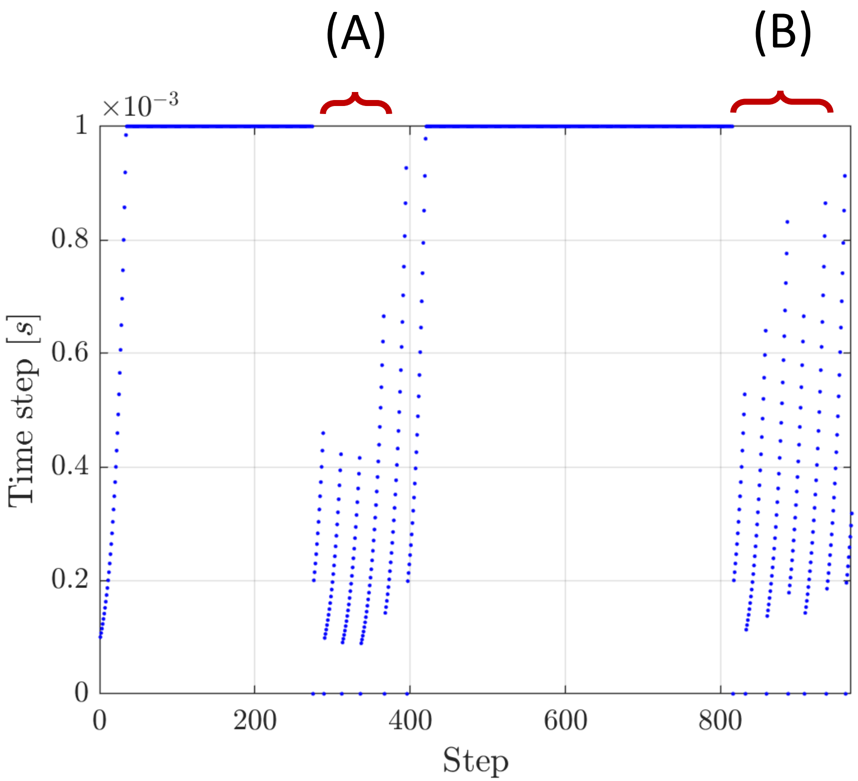


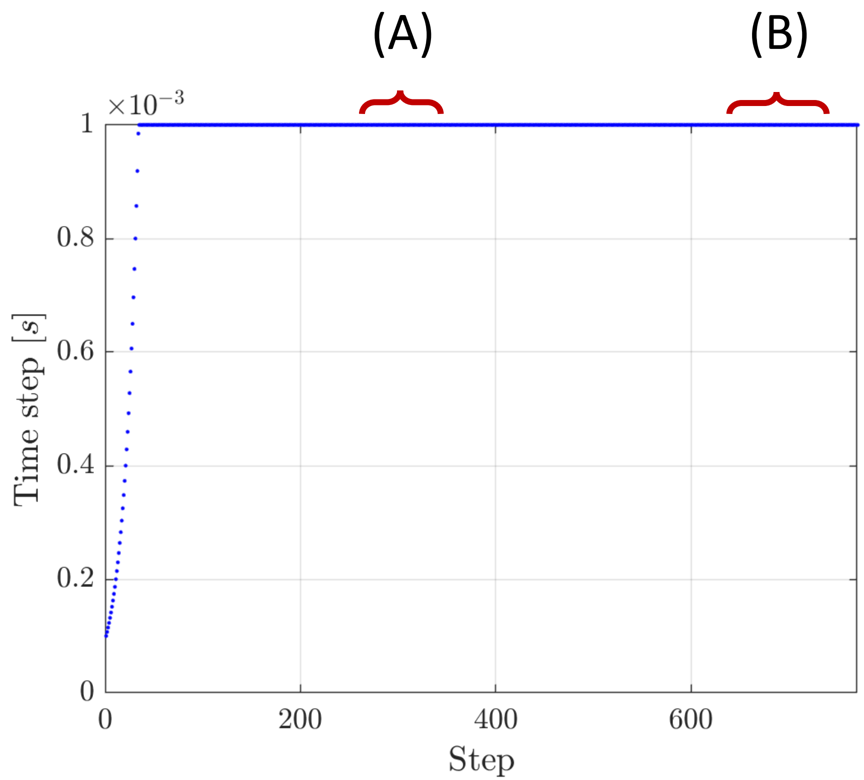
Study on the explicit time integration used in PFEM-2



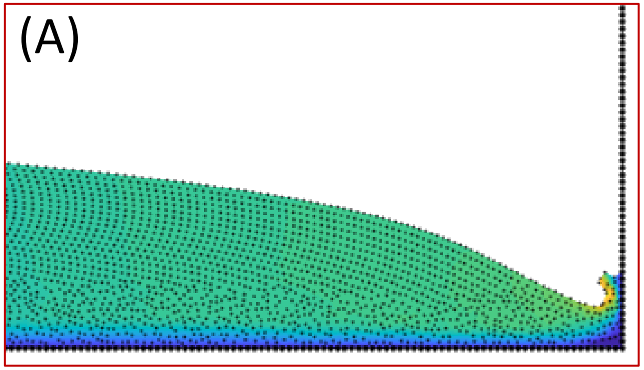


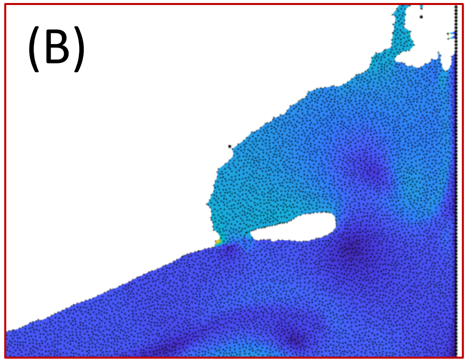
Reference $(h_{\text{char}} = 0.00365)$ **+ Remeshing (N-L loop) + XIVS integration**



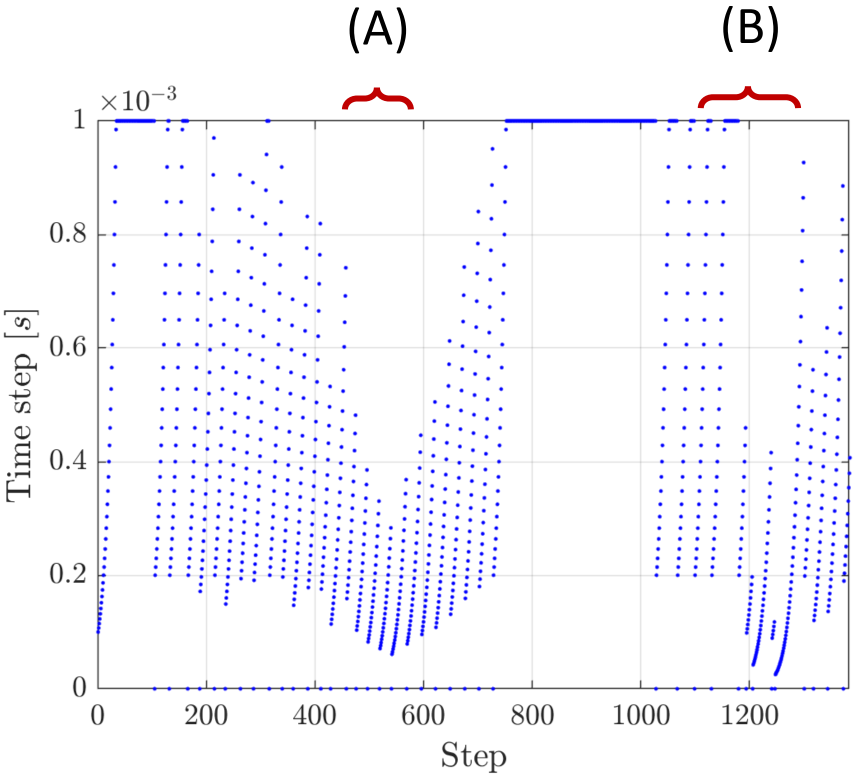


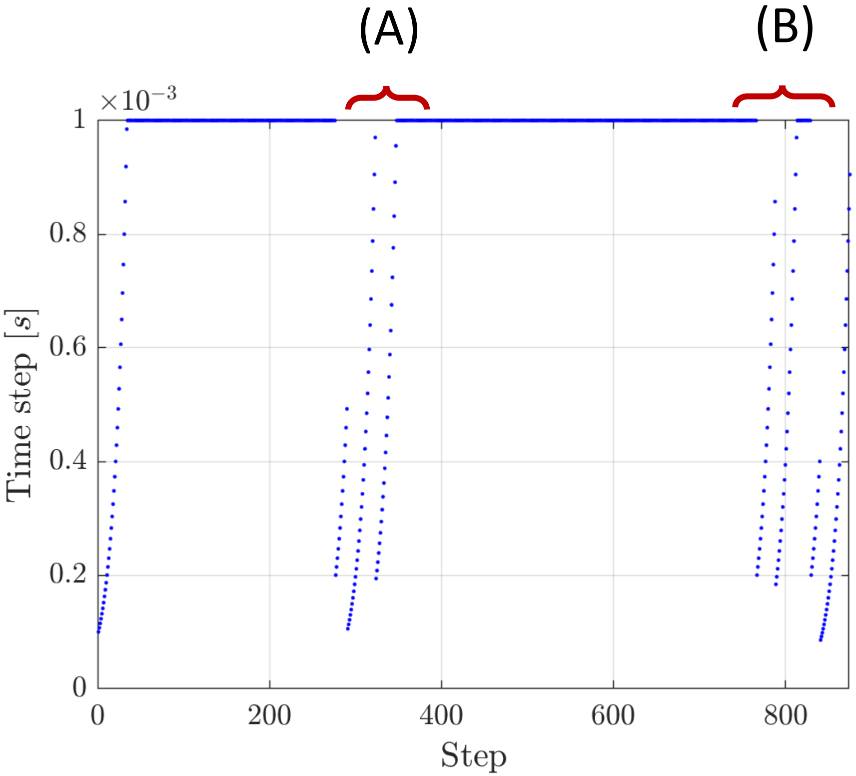
Study on the explicit time integration used in PFEM-2



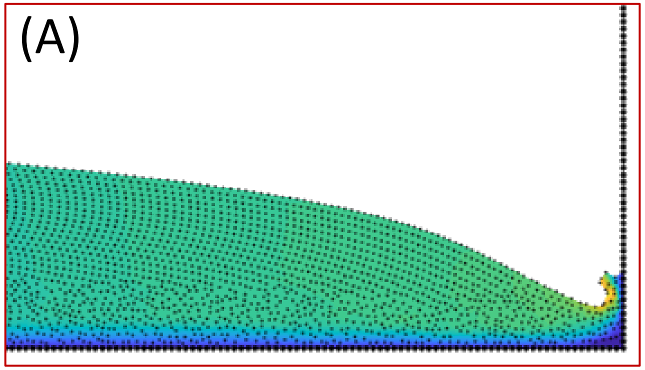


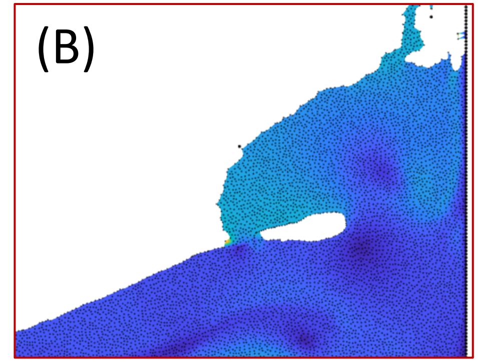
Reference $(h_{\text{char}} = 0.001825)$ **+ Remeshing (N-L loop) + XIVS integration**





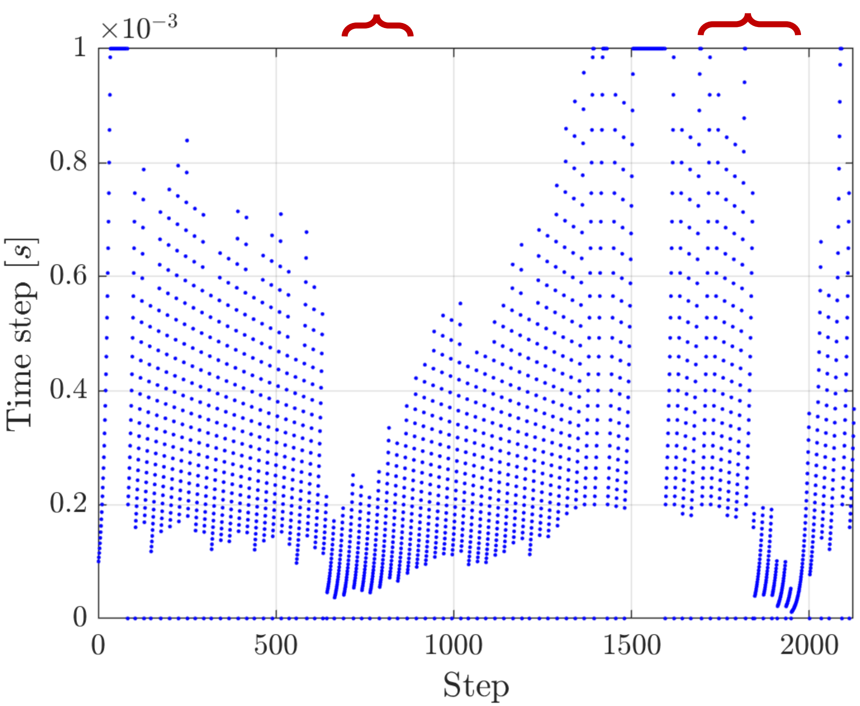
Study on the explicit time integration used in PFEM-2



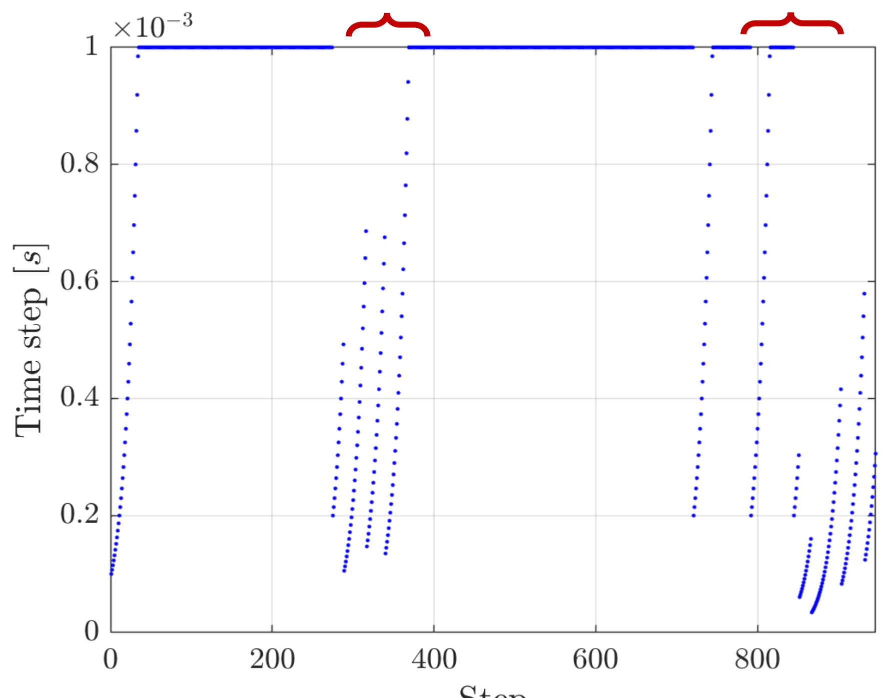


Reference $(h_{\mathrm{char}} = 0.0009125)$ **+ Remeshing (N-L loop) + XIVS integration**

(A) (B)



(A) (B)

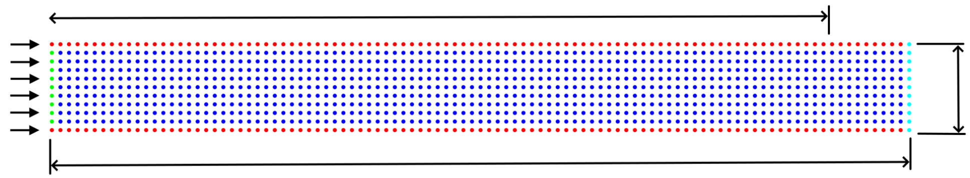


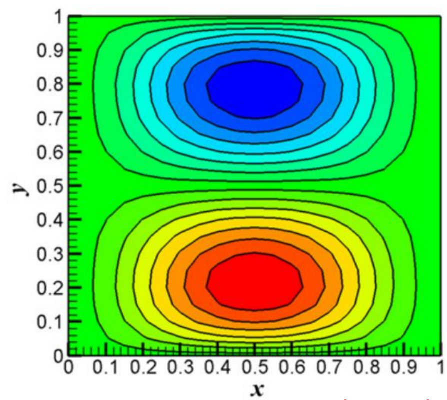
To do …

- Compare

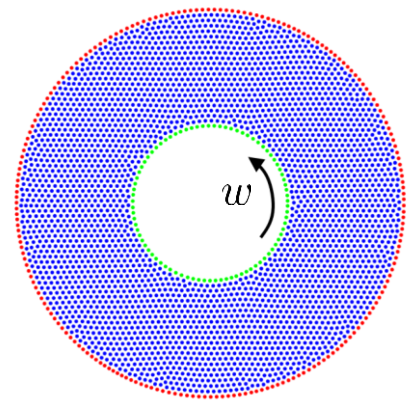
Integration \ Formulation	Velocity-Pressure	Position-Pressure
Backward-Euler		
α-Method		
+ XIVS (or Runge-Kutta)		

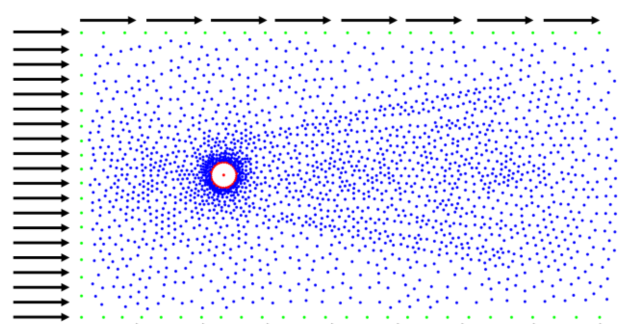
-- Using --





Oden & Jacquotte (1984)





To do …

- Compare
- Similar study for the Fractional Step
- C++ implementation (PFEM3D)
- Include thermal simulation (as Marti & Ryzhakov)*

* Marti, J., & Ryzhakov, P. (2020). An explicit/implicit Runge-Kutta-based PFEM model for the simulation of thermally coupled incompressible flows. *Computational Particle Mechanics*, *7*(1), 57-69.